

Buku Ajar Mata Kuliah Pengantar Mikroprosessor

**Oleh
Arief Wisaksono, Ir., MM.**



**Diterbitkan oleh
UMSIDA PRESS
Tahun 2019**

Buku Ajar
Pengantar Mikroprosesor

Penulis :
Arief Wisaksono, Ir., MM..

ISBN :
978-602-5914-99-7

Editor :
Septi Budi Sartika, M.Pd
M. Tanzil Multazam , S.H., M.Kn.

Copy Editor :
Fika Megawati, S.Pd., M.Pd.

Design Sampul dan Tata Letak :
Mochamad Nashrullah, S.Pd

Penerbit :
UMSIDA Press

Redaksi :
Universitas Muhammadiyah Sidoarjo
Jl. Mojopahit No 666B
Sidoarjo, Jawa Timur

Cetakan pertama, Agustus 2019

© Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dengan suatu
apapun tanpa ijin tertulis dari penerbit.

TINJAUAN MATA KULIAH

1. Deskripsi Singkat Mata Kuliah

Mahasiswa akan mempelajari tentang prinsip dan mekanisme kerja sistem mikroprosesor, bagaimana mikroprosesor melakukan operasi, kode program bahasa rakitan, dan eksekusi baris program. Akan dipelajari juga mengenai sinyal kendali dan antarmuka mikroprosesor dengan memori dan sistem I/O dalam pertukaran data. Diberikan juga materi mikrokontroler Arduino. Selain itu, diberikan juga materi tentang implementasi dari mikrokontroler Arduino

2. Kegunaan Mata Kuliah

Dengan mengikuti mata kuliah mikroprosesor mahasiswa akan mendapatkan pengetahuan mengenai rangkaian kontrol dalam sebuah industri, dan mahasiswa mampu menjelaskan dan menentukan indikator-indikator yang berpengaruh terhadap merancang sisten kontrol, serta mahasiswa dapat mengimplementasikan rancang bangun system control.

3. Standar Kompetensi

- a. Mahasiswa mampu untuk menjelaskan perbedaan mikrokontroller dengan Mikroprosesor, Arsitektur mikrokontroller.
- b. Mahasiswa dapat merencanakan dan membuat program mikrokontroller untuk berbagai aplikasi.
- c. Mahaiswa terampil dalam memilih piranti mikrokontroller untuk membuat berbagai aplikasi.
- d. Mahasiswa terampil merancang dan melaksanakan eksperimen dalam membangun aplikasi mikrokontroller.
- e. Mahasiswa mampu membuat program untuk berkomunikasi dengan komputer.

- f. Mahasiswa mampu bersikap kritis, inovatif dan berargumentasi dalam menyampaikan pendapat baik dalam bahasa lisan maupun tulisan.

3.2 Kompetensi Dasar

- a. Mengidentifikasi atribut-atribut sesuai konsep mikroprosesor dan Mikrokontroler
- b. Mampu melakukan merancang dan melaksanakan eksperimen dalam membangun aplikasi mikrokontroller.
- c. Mampu menerapkan merencanakan dan membuat program mikrokontroller untuk berbagai aplikasi.

3.3 Indikator

- a. Mahasiswa dapat menjelaskan konsep mikroprosesor dan Mikrokontroler tanpa melihat *handout*.
- b. Mahasiswa mampu memberikan contoh atribut dari mikrokontrolernya.
- c. Mahasiswa dapat menjelaskan bagian-bagian arduino.
- d. Mahasiswa dapat memprogram arduino
- e. Mahasiswa mampu merangkai Aplikasi rangkaian arduino

4. Petunjuk Bagi Mahasiswa

Untuk membantu mahasiswa dalam mempelajari buku ajar pengantar mikroprosesor ini, dapat dilakukan beberapa tahapan sebagai berikut:

- 1. Bacalah dan pamilah dengan teliti dari isi dan makna dari setiap bab. Kerjakan soal yang terdapat pada akhir setiap bab, kemudian lanjutkan ke bab selanjutnya.

2. Buku ajar ini disusun saling berkaitan antar bab, sehingga mahasiswa dalam mempelajari bab berikutnya harus memahami bab sebelumnya dengan baik.
3. Untuk membelajarkan secara mandiri, disarankan mahasiswa dapat melengkapi dengan referensi lain yaitu buku teks yang berbeda namun materi yang disampaikan sama serta beberapa jurnal hasil penelitian.

KATA PENGANTAR

Dengan mengucapkan puji syukur Allhamdullillah, atas berkat rahmat Allah SWT, kami dapat menyelesaikan buku ajar dengan judul **“Pengantar Mikroprosesor”**. Kami selaku penyusun mengucapkan terima kasih kepada semua pihak yang telah membantu kami selama proses pelaksanaan penyusunan sampai dengan terselesainya buku ajar ini. Kami menyadari, dalam buku ajar yang kami susun masih banyak kekurangan, sehingga kami berharap pembaca dan pengguna dapat memberikan masukan/ kritik yang sifatnya membangun. Semoga apa yang kami hasilkan ini dapat memberikan manfaat bagi pembaca dan mahasiswa.

Sidoarjo, Juni 2019

Penyusun

DAFTAR ISI

Cover	1
Tinjauan Mata Kuliah	3
Kata Pengantar	5
Daftar Isi	6

BAB 1 PENGENALAN MIKROPROSESOR DAN KOMPUTER

1 CPU dan Mikrokomputer	
2 Pendahuluan	9
3 Pengertian Dasar Mikroprosesor	10
4 Kemasan Mikroprosesor	13
5 Struktur Mikroprosesor	15
6 Sistem Bus pada Mikroprosesor	16
7 Perkembangan Mikroprosesor	19

BAB 2 Bahasa assembly dengan debug

2. Pendahuluan	27
3. Pengertian segment dan offset	28
4. Interrupt	28
5. Bagian – bagian dari program assembler	29
6. Assembly dengan debug.com	31

BAB 3 Mikrokontroler

4. Arsitektur mikrokontroler	68
5. Rangkaian clock dan reset	76
6. Power suplay	79

BAB 4 Arduino Mikrokomputer

4.1 Pengertian Arduino	83
4.2 Kelebihan Arduino	86
4.3. Bagian – Bagian Arduino	89

BAB V : Implementasi dari Rangkaian Arduino

5.1 Pendeteksi Kebocoran Gas Berbasis Arduino uno	94
5.2 Sensor Intensitas Cahaya	97
5.3 Mengendalikan LED Menggunakan Perintah Suara	99
5.4 Robot Penghalang Rintangan	102
5.5 Sensor Parkir.. Mobil	110

BAB I

PENGENALAN MIKROPROSESOR DAN KOMPUTER

Mata Kuliah : Mikroprocessor

Capaian Pembelajaran :

Memahami arsitektur dasar mikroprosesor , mekanisme proses, dan komponen logika penyusunnya

Kemampuan Akhir yang Diharapkan :

Mahasiswa mampu memahami (menjelaskan) arsitektur dasar mikroprosesor,

Indikator :

- A. [C2] Mahasiswa akan mampu menjelaskan arsitektur mikroprosesor
- B. [C2] Mahasiswa akan mampu menjelaskan bus yang terdapat dalam mikrokomputer
- C. [C2] Mahasiswa akan mampu menjelaskan I/O yang terdapat dalam mikrokomputer
- D. [C4] Mahasiswa akan mampu mengidentifikasi peripheral yang ada di satu sistem mikrokomputer dan antarmukanya
- E. .[C2] Mahasiswa akan mampu menjelaskan perbedaan antarmuka serial dan parallel

1. CPU dan Mikrokomputer

1.1. PENDAHULUAN

Rangkaian terintegrasi universal yang sangat murah dapat diprogram dan digunakan di banyak bidang perangkat elektronik, atau di mana pun diperlukan, teknologi ini telah berkembang sangat pesat. Sebuah mikrokontroler (disebut sebagai μC , uC atau MCU) adalah sirkuit terintegrasi tunggal yang terdiri dari inti prosesor, memori, dan terminal input / output yang dapat diprogram berfungsi sebagai komputer mini. Memori program dalam bentuk (NOR) flash atau OTP ROM juga sering disertakan dalam chip, serta jumlah RAM yang biasanya kecil. Mikrokontroler dirancang untuk aplikasi tertanam, berbeda dengan mikroprosesor yang digunakan di komputer pribadi. Mikrokontroler banyak digunakan dalam produk dan perangkat yang dikendalikan secara otomatis, seperti sistem kontrol mesin mobil, perangkat medis implan, kontrol jarak jauh, mesin kantor, peralatan, peralatan listrik, mainan dan sistem tertanam lainnya. Dengan mengurangi ukuran dan biaya dibandingkan dengan desain yang menggunakan mikroprosesor, memori, dan perangkat input / output yang terpisah, mikrokontroler membuatnya lebih ekonomis untuk mengontrol secara digital bahkan lebih banyak perangkat dan proses.

Mikrokontroler sinyal campuran adalah umum, mengintegrasikan komponen analog yang diperlukan untuk mengontrol sistem elektronik non-digital.

Beberapa mikrokontroler dapat menggunakan kata-kata empat-bit dan beroperasi pada frekuensi clock rate serendah 4 kHz, untuk konsumsi daya yang rendah (mill watts atau microwatts). Mereka umumnya akan

memiliki kemampuan untuk mempertahankan fungsionalitas sambil menunggu acara seperti tekan tombol atau interupsi lainnya; konsumsi daya saat tidur (jam CPU dan sebagian besar perangkat mati) mungkin hanya nanowatt, membuat banyak dari mereka sangat cocok untuk aplikasi baterai tahan lama.

Mikrokontroler lain mungkin melayani peran penting-kinerja, di mana mereka mungkin perlu bertindak lebih seperti prosesor sinyal digital (DSP), dengan kecepatan clock yang lebih tinggi dan konsumsi daya.

1.1. Pengertian Dasar Mikroprosesor

Komponen utama sebuah sistem *Mikroprosesor* tersusun dari lima unit pokok: unit mikroprosesor atau *Microprocessor Unit* (MPU) atau CPU, unit memori baca atau *Read Only Memory* (ROM), unit memori baca tulis atau *Read Write Memory* (RWM), unit masukan keluaran terprogram atau *Programmable Input Output*(PIO) dan unit detak/*Clock*.

Ada tiga jenis arsitektur *Mikroprosesor*:

1. Arsitektur I/O Terisolasi

Mikroprosesor dengan arsitektur I/O Terisolasi menggunakan disain pengalamatan atau pemetaan I/O terpisah atau terisolasi dengan pengalamatan atau pemetaan memori. Pengalamatan I/O menggunakan sebagian dari jumlah saluran alamat (*Address Buss*) sedangkan pengalamatan memori menggunakan semua saluran alamat (*Address Buss*).

Metode I/O terisolasi menggunakan akumulator pada CPU untuk menerima informasi dari I/O atau mengeluarkan informasi ke bus I/O selama operasi *Input Output*. Tidak ada *Register* lain selain akumulator yang terpakai untuk akses I/O. Metode I/O Terisolasi disebut juga dengan I/O akumulator. Konsep ini memiliki pengaruh penting pada program komputer yaitu:

- Instruksi yang digunakan hanya dua kode operasi yaitu IN dan OUT
- Informasi/data yang ada pada akumulator harus dialihkan pada suatu lokasi penyimpanan sementara sebelum ada operasi I/O berikutnya
- Perlu ada tambahan instruksi pada program pengalihan data/informasi pada akumulator

Keuntungan metode I/O terisolasi:

- Komputer dapat mengalihkan informasi/data ke atau dari CPU tanpa menggunakan memori. Alamat atau lokasi memori untuk rangkaian memori bukan untuk operasi I/O
- Lokasi memori tidak berkurang oleh sel-sel I/O Instruksi I/O lebih pendek sehingga dapat dengan mudah dibedakan dari instruksi memori
- Pengalamatan I/O menjadi lebih pendek dan perangkat keras untuk pengkodean alamat lebih sederhana.

Kerugian metode I/O terisolasi:

Lebih banyak menggunakan penyemat pengendalian pada *Mikroprosesornya*. *Mikroprosesor* buatan *Intel* dan *Mikroprosesor* buatan *Zilog* menggunakan arsitektur I/O Terisolasi.

2. Arsitektur I/O Terpetakan dalam Memori

Mikroprosesor dengan arsitektur I/O terpetakan dalam memori menyatukan sel-sel I/O dalam pengalamatan yang bersama dengan sel-sel memori. I/O yang terpetakan dalam memori menunjukkan penggunaan instruksi tipe memori untuk mengakses alat-alat I/O.

I/O yang dipetakan dalam memori memungkinkan CPU menggunakan instruksi yang sama untuk alih memori seperti yang digunakan untuk alih I/O. Sebuah pintu I/O diperlakukan seperti sebuah lokasi memori. Keuntungan sistim ini adalah instruksi yang dipakai untuk pembacaan dan penulisan memori dapat digunakan untuk memasukkan dan mengeluarkan data pada I/O.

Kerugiannya pertama tiap satu pintu I/O mengurangi satu lokasi memori yang tersedia. Kedua alamat lokasi I/O memerlukan 16 bit saluran. Ketiga instruksi I/O yang dipetakan dalam memori lebih lama dari instruksi I/O terisolasi.

3. Arsitektur *Harvard*

Arsitektur *Harvard* menggunakan disain yang hampir sama dengan arsitektur I/O terisolasi. Perbedaannya pada arsitektur harvard antara memori program dan memori data dipisahkan atau diisolasi.

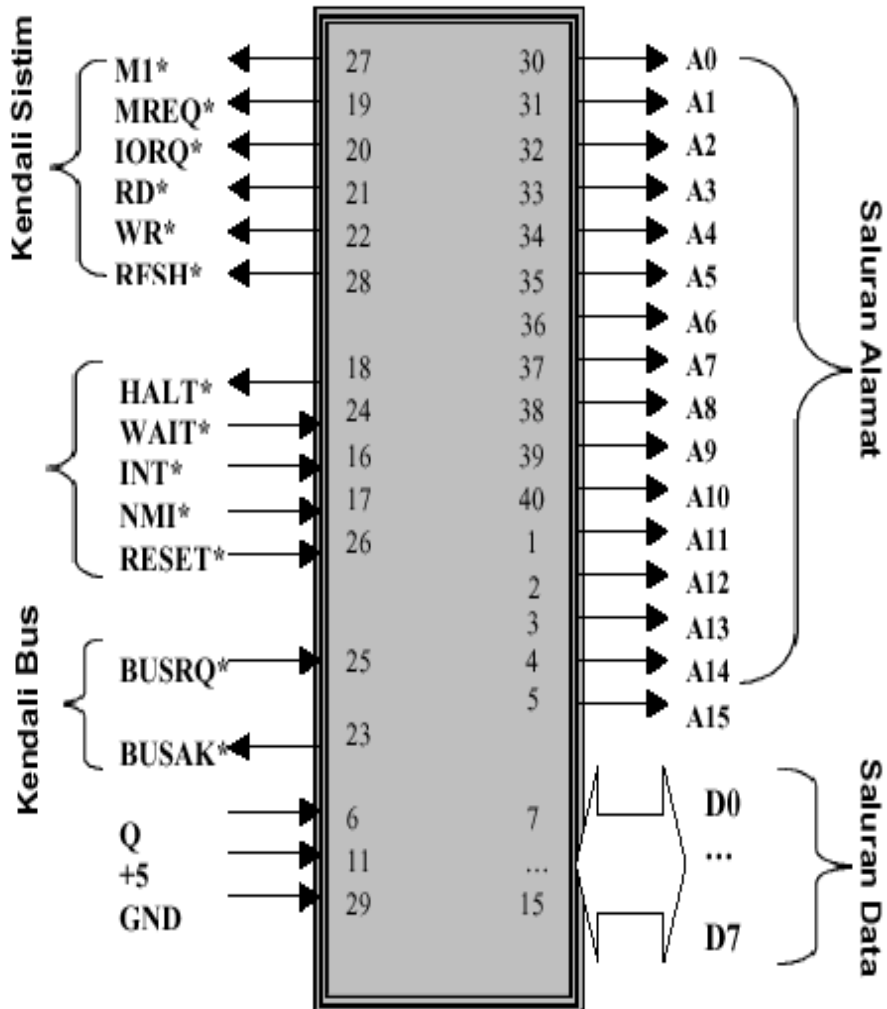
Pemisahan antara memori program dan memori data menggunakan perintah akses memori yang berbeda. *Harvard* arsitektur ditinjau dari kemampuan jumlah memori lebih menguntungkan.

1.3. Kemasan Mikroprosesor

Ada empat jenis bentuk kemasan *Mikroprosesor*:

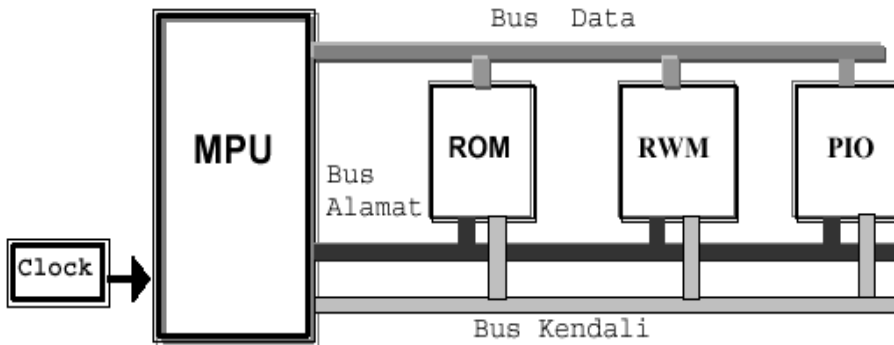
- PDIP: *Pastic Dual Inline Package*
- PLCC: *Plastic J-Lieded Chip Carrier*
- TQFP: *Plastic Gull Wing Quad Flat Package*
- SOIC: *Plastic Gull-wing Small Outline.*

Feature Kasus pada Zilog Z-80 CPU



Gambar 4. Susunan dan Konfigurasi Pin Z-80 CPU

1.4. Struktur Mikroprosesor



Gambar Blok Diagram Sistem Mikroprosesor

MPU adalah sebuah CPU yang tersusun dari tiga bagian pokok yaitu:

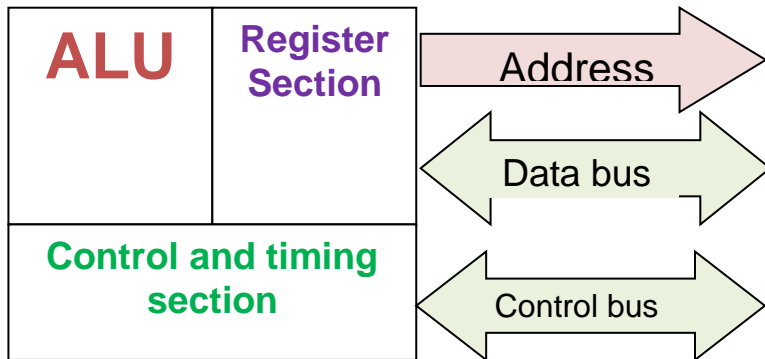
- *Control Unit* (CU)
- *Arithmetic Logic Unit* (ALU)
- *Register Unit* (RU)

Sebagai CPU, MPU bekerja dan melakukan fungsi dasar yaitu fungsi Logika dan Aritmetika. Fungsi Logika antara lain fungsi AND, OR, XOR, CPL, dan NEG. Sedangkan fungsi Aritmetika antara lain: ADD, SUB, ADC, SBC, INC, dan DEC. Disamping fungsi pengolahan Aritmetika dan Logika MPU juga melakukan fungsi pengalihan data dengan menggunakan perintah MOV, atau LOAD, EXCHANGE, PUSH, dan POP. Untuk menyimpan program dan data yang digunakan pada sistem *Mikroprosesor* harus dilengkapi dengan Memori.

Jadi memori mutlak diperlukan dalam *Sistim Mikroprosesor*. Tanpa ada memori *Sistim Mikroprosesor* tidak dapat bekerja terutama memori program dalam ROM. I/O unit dipersiapkan untuk menghubungkan MPU dengan alat-alat *input-output* luar seperti *Keyboard, Monitor, Printer, Mouse*, dan sebagainya.

1.5. Sistim Bus pada Mikroprosesor

Mikroprosesor berkomunikasi dengan unit memori, unit I/O menggunakan saluran yang disebut dengan *BUS* yang digunakan untuk sambungan dengan komponen-komponen pendukung system mikroprosesor. Secara blok diagram bisa digambarkan seperti pada gambar dibawah



Saluran – saluran tersebut dikelompokkan sebagai berikut:

- o Bus saluranAlamat (Address Bus)
- o Bus saluran Data (Data Bus)
- o Bus saluranKendali (Control Bus)

Address Bus

Saluran alamat (address bus) diperlukan untuk menentukan suatu lokasi alamat memori maupun lokasi alamat dari perangkat input / output (Input Output Interface) yang selalu digunakan dalam suatu system mikroprosesor itu sendiri, sebab jumlah saluran yang ada langsung menentukan banyaknya alamat memori (kapasitas memori) yang dapat ditanganinya.

Data Bus

Data bus (saluran data) diperlukan sebagai jalan masukan atau keluaran data yang berfungsi sebagai instruksi atau penyerta instruksi antara perangkat ingatan dan perangkat I/O dengan CPU. Untuk mikroprosesor Z80 memiliki 8 buah saluran data (D0D7) atau 8 bit data. Sifat dari saluran data adalah dua arah (bidirectional), jadi arahnya dapat bolak-balik antar mikroprosesor dengan perangkat ingatan maupun perangkat masukan keluaran.

Control Bus

Bus Kendali adalah seperangkat bit pengendali yang berfungsi mengatur:

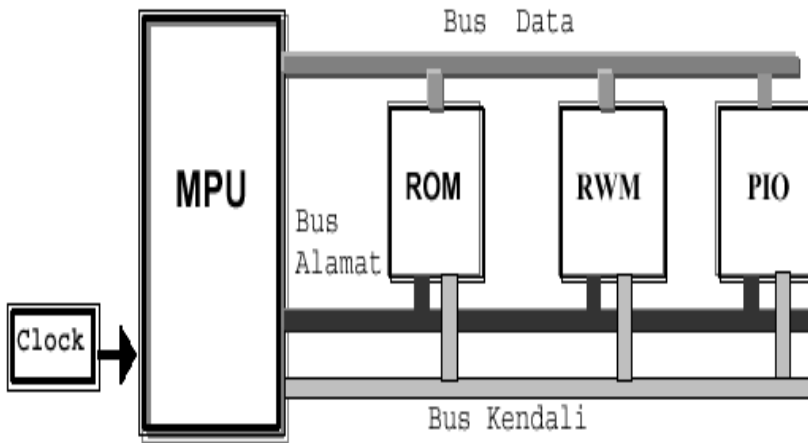
- (1) Penyerempakan memori,
- (2) Penyerempakan I/O,
- (3) Penjadualan MPU, Interupsi, DMA ,
- (4) Pembentuk *Clock*, dan *Reset*.

Saluran kendali (Control Bus) digunakan untuk melakukan pengendalian terhadap komponen-komponen pendukung dalam system mikroprosesor. Semua aktifitas lalu lintas data maupun pengambilan instruksi perangkat ingatan, perangkat keluaran masukan dan lain-lainnya diatur dan dikendalikan lewat saluran kendali tersebut.

Tabel Sistem Bus

Nama Buss	Sifat	Arah Data dari CPU	Jumlah Saluran
Bus Data	Dua arah	Masuk dan Keluar	8 bit
Bus Alamat	Satu arah	Keluar	16 bit
Bus Kendali	Satu arah	Masuk dan Keluar	10–12 bit

Alih data diantara MPU dengan komponen luar berlangsung pada Bus Data. *Mikroprosesor* standar memiliki saluran bus data 8 bit dua arah artinya alih data atau informasi berlangsung pada 8 saluran paralel dari MPU ke unit lain diluar MPU atau dari unit lain di luar ke MPU.



Gambar Blok Diagram Sistem Bus

1.6. Perkembangan Mikroprosesor

Mikroprosesor sebagai komponen utama dalam *Sistim*

Mikroprosesor dapat dikelompokkan menurut:

- (a) Teknologi yang digunakan;
- (b) Jumlah Bit Data;
- (c) Kemampuan atau *Karakteristik Mikroprosesor*.

Tabel 2 menunjukkan pengelompokan perkembangan *Mikroprosesor*.

Tabel Pengelompokan *Mikroprosesor*

No	Jumlah Bit Data	Contoh MP	Jenis Teknologi	Tahun Buatan	Clock (MHz)	Karakteristik		
						Kecepatan	Daya	Kepadatan (10^6)
1	4 bit	4004 4040 8008	PMOS			Rendah	Sedang	Tinggi
2	8 bit	8080 8085 8088 6800 6802 6809 6801 6805 Z80	NMOS	1979	8	Rendah Sedang	Sedang	0,029
3	16 bit	8086 80186 80188 80286 80288 68000 Z8000 9900	HMOS NMOS	1978 1982 1982	10 16 16	Rendah Sedang	Sedang	0,029 0,029 0,134
4	32 bit	80386DX 80386SX 80486DX 80486SX 68020 NS32032 WE3200	NMOS	1985 1988 1989 1991	16-33 16-33 25-50 16-33	Rendah Sedang	Sedang	0,275 0,275 1,2 1,185
5	32 bit 64	P54VRT Pent Pro Pent II Pent III Itanium	NMOS	1994 1995 1998 1999 2000	75-150 150-200 233-450 450-600 533	Rendah Sedang	Sedang	3,1 5,5 7,5 9,5

Disamping teknologi PMOS (*Metal-Oxide Semiconductor* kanal P) dan teknologi NMOS (*Metal-Oxide Semiconductor* kanal N) yang paling banyak digunakan sebagai teknologi pembuatan mikroprosesor masih ada teknologi lain yaitu:

- Teknologi CMOS (*Complementary Metal-Oxide Semiconductor*)
- Teknologi CMOS-SOS (*teknologi CMOS menggunakan substrat Sphir- Silicon-On-Sapphire*)
- Teknologi Bipolar jenis ECL (*Emitter-Coupled-Logic*)
- Teknologi Bipolar jenis *Schottky*
- Teknologi Bipolar jenis I²L (*Integrated-Injection-Logic*)
- Mengingat makin banyaknya macam dan jenis *Mikroprosesor* yang ada sampai saat ini, maka sebagai pemakai kita perlu menentukan macam komponen yang paling sesuai dengan keperluan kita. Dalam memilih komponen *Mikroprosesor* beberapa hal pertimbangan perlu dikaji misalnya:
- Pertimbangan *Sistim* meliputi karakteristik sistim, jumlah *Supplier*, harga, dan ketersediaan.
- Pertimbangan *Hardware* meliputi jumlah bit data, macam, kemampuan dan waktu instruksi dan macam bahasa.

Clock

Merupakan bagian dari *Sistim Mikroprosesor* yang mengatur denyut kerja MPU. Sehingga *Frekuensi Clock* berkaitan dengan kecepatan

kerja komputer. Beberapa jenis MPU ada yang menggunakan detak sistim tunggal dan ada juga sistim ganda (*dual fase*). Detak dapat dibangkitkan menggunakan sistim diskrit atau IC khusus. *Intel* memperkenalkan IC 8224 untuk penggerak detak.

Pengendalian Sistem Mikroprosesor

MPU dalam suatu sistem mikroprosesor dalam fungsinya sebagai pengendali sistem bekerja sebagai:

- Pengendali sistim
- Pengendali bus/saluran
- Dikendalikan oleh alat luar.

Pada Tabel berikut digambarkan *Ekivalensi* sinyal-sinyal kendali beberapa jenis *Mikroprosesor*.

Tabel Sinyal Kendali *Mikroprosesor*

Sinyal Pengendali	Jenis Mikroprosesor				
	8080/8228	8085	Z-80	6800	6502
PENGENDALI SISTIM	SYNC	-	M1	-	SYNC
	MEMR	RD&IO/M	RD&MEMRQ	R/WQ2	R/W&Q2
	MEMW	WR&IO/M	WR&MEMRO	R/W&Q2	R/W&Q2
	I/ORD	RD&IO/M	RD&IORQ	R/W&Q2	R/W&Q2
	I/OWR	WR&IO/M	WR&IORQ	R/W&Q2	R/W&Q2
PENGENDALI MPU	-	-	HALT	-	-
	READY	READY	WAIT	-	RDY
	INT	INTR	INT	IRO	IRO
	-	TRAP	NMI	NMI	NMI
	RESET	RESET	RESET	RESET	RESET
PENGENDALI BUS	HOLD	HOLD	BUSRQ	HALT	RDY
	HLDA	HLDA	BUSAK	BA&VMA	-

Penyerempakan memori dan penyerempakan I/O pada pokoknya analogis. Digunakan prosedur jabat tangan. Dalam operasi “baca” suatu status sinyal “siap” (*Ready*) akan menunjukkan tersedianya data. Kemudian data dialihkan ke bus data. Pada beberapa alat I/O dibangkitkan suatu sinyal “pengakuan” (*acknowledge*) untuk memberitahukan penerimaan data. Pembangkitan sinyal pengakuan ini menggunakan sistim tak serempak (*Asinkron*). Pada sistem sinkron tidak diperlukan adanya pembangkitan sinyal pengakuan.

Ciri dari sistem sinkron adalah:

- Kecepatan yang lebih tinggi
- Jumlah saluran bus pengendali lebih sedikit
- Pembatasan kecepatan pada alat-alat I/O.
- Pada sistem asinkron tercirikan adanya:
- Jumlah saluran bus pengendali lebih banyak

- Memungkinkan penggunaan piranti berkecepatan berbeda dalam satu sistem yang sama.

2. PENUTUP

2.1. Tes

Tes diberikan saat ujian tengah semester, soal diberikan lebih bersifat menerapkan atau mempraktekkan teori yang ada pada obyek kehidupan nyata. Bentuk ujian berifat jurnal dan project (Discovery learning Project Base Learning).

2.2. Tindak Lanjut

Soal UTS akan dibahas pada pertemuan selanjutnya dan bersifat diskusi serta tanya jawab dan dikembangkan pada Ujian akhir semester.

Referensi:

1. Frank Vahid and Tony Givargis, "Embedded System Design: A Unified Hardware/Software Approach", 1999
2. Ken Arnold, "Embedded Controller Hardware Design", 2000
3. Buku-buku mikrokontroler lainnya

BAB II

BahasaAsembly

Mata Kuliah : Mikro Prosessor

Capaian Pembelajaran :

Mahasiswa Mampu menerapkan dan mengembangkan Set instruksi bahasa Asmbley , dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi

Kemampuan Akhir yang diharapkan :

Mahasiswa memahami set instruksi dan sifatnya. dalam Sistem Komputer

Indikator :

- a) Mahasiswa akan mampu menjelaskan instruksi bahasa asembly
- b) Mahasiswa akan mampu menuliskan set i
- c) Mahasiswa akan mampu menstruksi

2. Bahasa asembly dengan debug

2.1. Pendahuluan

Bahasa Assembly adalah bahasa yang memudahkan pemahaman bagian computer yang paling rendah, mendekati mesin. Setiap bahasa assembly secara langsung dipengaruhi oleh set intruksi mesin computer dan arsitektur perangkat keras. Misalnya, bahasa assembly IBM-PC adalah bahasa assembly yang mengacu pada instruksi-instruksi yang dikenali oleh keluarga mikroprosesor intel 8086-80486. Assembler juga dikenal sebagai bahasa pemrograman yang mengonversi kode program sumber ke dalam bahasa mesin. Terdapat dua assembler yang dikenal baik untuk IBM-PC , yaitu MASM (Microsoft Assembler) dan TASM (Turbo Assembler). Bahasa assembly disebut bahasa level-bawah karena dalam struktur dan fungsi dekat dengan bahasa mesin. Sebaliknya, bahasa tingkat tinggi seperti Pascal, Basic, Fortran, dan Cobol mempunyai perintah-perintah yang andal yang diterjemahkan ke dalam berbagai instruksi mesin oleh compiler.

Kelebihan

- Hasil program memiliki tingkat kecepatan yang tinggi
- Ukuran dari program lebih kecil
- Sangat mudah untuk mengakses sistem komputer

2.2. Pengertian segment dan offset

Segment dan offset merupakan angka 16 bit (dipresentasikan dalam bahasa hexa) yang menunjukan suatu alamat tertentu di

memory komputer. Pasangan segment : offset ini disebut juga alamat relatif. Selain alamat relatif, terdapat juga alamat mutlak berupa angka 20 bit (juga dipresentasikan dalam bil hexa). Alamat mutlak ini dapat dihitung dengan mengalihkan segment dengan 10 hexa dan ditambahkan dengan offset.

2.3. Interrupt

Interrupt merupakan permintaan kepada microprocessor untuk melakukan suatu perintah. Ketika terjadi permintaan interupsi, microprocessor akan mengeksekusi interrupt handler, yaitu suatu program yang melayani interupsi. Setiap interrupt handler itu memiliki alamat masing-masing yang disimpan dalam bentuk array yang masing-masing terdiri dari 4 byte. Array ini disebut vektor interupsi.

2.4. Bagian – bagian dari program assembler

- Label

Label merupakan suatu simbol yang didefinisikan sendiri oleh pembuat program untuk menandai lokasi memori pada area program. Simbol dan label adalah dua hal yang berbeda. Simbol tidak menggunakan titik dua, sedangkan label harus diakhiri dengan titik dua.

Contoh :

EQU 500 ; “PAR” Menunjukan suatu symbol

; dari nilai 500

MULAI :

```
MOV A, #0FFh ; pada label; "Mulai" nilai 0FFh  
; dipindahkan ke Akumulator
```

Dalam satu baris hanya ada satu label, pada umum nya Assembler membatasi jumlah karakter yang bisa digunakan hingga 31 karakter.

- Mnemonik

Mnemonic instruksi atau pengarah Assembler dimasukan dalam "Mnemonic field" yang mengikuti "label mnemonic". Mnemonic instruksi misalnya ADD, MOV, INC dan lain-lain. Sedangkan pengarah Assembler misalnya ORG, EQU, DB dan lain-lain.

- Operand

Operand ditulis setelah mnemonic, bisa berupa alamat atau data yang digunakan instruksi yang bersangkutan.

Contoh :

```
MOV A, #20h ; Adan #20h adalah operand
```

LAGI:

```
JNB LAGI ; LAGI adalah operand
```

- Komentar

Komentar harus diawali dengan titik koma. Sub rutindari bagian besar program yang mengerjakan suatu operasi biasanya diawali

dengan blok komentar yang menjelaskan fungsi sub rutin atau bagian besar program tersebut.

- End

Petunjuk END merupakan kode perintah terakhir yang menunjukkan batas akhir dari proses Assembly.

Penulisan perintah:

MOV [operand A],[operand B]

Dengan ketentuan operand A merupakan register, variabel, lokasi memori dan ketentuan isi operand B berupa register, variabel, lokasi memori ataupun bilangan.

Operand B merupakan bilangan asal yang akan diisikan ke operand A, dengan kata lain *operand A* merupakan *tujuan pengisian* atau *penduplikatan* dari *operand B*.

Contoh:

MOV AH, AL

Operand A dari perintah diatas adalah register AH

Operand B dari perintah diatas adalah register AL

Hal yang dilakukan dari perintah diatas adalah menduplikatkan isi register AL ke register AH

MOV AH,02

Operand A dari perintah diatas adalah register AH

Operand B dari perintah diatas adalah bilangan 02

Hal yang dilakukan dari perintah diatas adalah memasukkan 02 ke register AH

2.5. Asembly dengan debug.com

Pembelajaran assembly pada level dasar yakni belajar assembly dengan debug. Adapun peralatan yang dibutuhkan adalah :

1. Komputer
2. Aplikasi Debug.com

Berikut urutan proses yang dilalui.

1. Aktifkan komputer
2. Aktifkan terminal Dos operation atau tekan Cmd
3. Masuk ke direktori system32 atau tekan `cd\windows\system32`
4. Ketik debug
5. Selanjutnya tulus setiap contoh dibawah.
6. Berikut contoh instruksi

Contoh Perintah Debug

- A (Assemble/Address) :
Memulai pembuatan program
rakitan dengan Debug
 - Syntax : A <offset>
 - Offset Merupakan segment
memory program dimulai,
biasanya pada offset 100h
 - Contoh :
 - C:\debug
- A100

Contoh Perintah Debug

- C
(Singkatan dari Compare) :
berfungsi untuk
membandingkan isi address
memory tertentu dengan isi
address memory lainnya.
 - Syntax : C <segment><L
panjang><segment2>
 - Contoh-1 : -C 0100 L10 0200
 - Berarti memulai offset 0100 sebanyak 10
Hexa untuk membandingkan dengan
offset 0200
 - Contoh-2 : -C 0100 L8 0200

Contoh Perintah Debug

- D (Dump) : Menampilkan data yang tersimpan pada segment memory dan offset
 - Syntax : D <segment:offset> [n] Akan ditampilkan datanya
 - Contoh : -D CS:80
(Menampilkan data yang tersimpan pada segment memori CS:0080 sampai:0160)
 - Hasil instruksi dibagi 3, yaitu :
 - Kiri [n] Tampilan segment memori yang memiliki komposisi Segment:offset
 - Tengah [n] Angka dalam format heksa merupakan isi dari segment memori
 - Kanan [n] Kode ASCII dari terjemahan angka Hexa

Contoh Perintah Debug

C:\>debug

-D CS:80

137C:0080 00 0D 42 4C 41 53 54 45-52 3D 41 32 32 30 20 49

..BLASTER=A220 I

137C:0090 35 20 44 31 20 50 33 33-30 20 54 33 0D 74 2E 65 5

D1 P330 T3.t.exe

137C:00A0 78 65 0D 00 00 00 00 00-00 00 00 00 00 00 00 00

137C:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

137C:00C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ...

137C:00D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

137C:00E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

137C:00F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

-

- Beberapa parameter yang bisa digunakan dalam instruksi Dump:
- L (length/panjang) : data sepanjang 2 byte, bila parameter “L” tidak diberikan otomatis akan ditampilkan 128 byte
Contoh : -D 0100L2
- Segment awal –segment akhir. Contoh : -D 0100 01FF
- Segment segment:offset. Contoh : -D FFFF:0000
- Segment segment : offset sampai segment ; offset. Contoh : -D F000:E000

Contoh Perintah Debug

- **E (Enter/edit) : Mengisi/ data mengubah pada memori yang ditunjuk segment & Offset**
 - Syntax : E <segment:offset>
 - Contoh : -E 0100 137C:0100 BB_
 - Untuk mengubah nilai pada karakter heksadesimal, masukkan nilai baru
137C:0100 BB_ 35
 - Pada contoh ini nilai BB pada offset 100 diubah menjadi 35
 - Ulangi proses berikut, jika menekan tombol space 3 kali, debug akan menampilkan 3 offset selanjutnya 137C:0100 BB_ 35
01 00 B6
 - Untuk mengubah byte pada offset 100, tekan tombol *hyphen* sebanyak 3 kali dan debug akan

kembali ke offset 100

Contoh Perintah Debug

- G (Go) : menjalankan/eksekusi program
- H (Hex) : Melaksanakan penjumlahan dan pengurangan 2 bilangan Hexa.
 - Syntax : H <operand1><operand2>
 - Contoh 1 : -H 3 4 0007 FFFF
 - H 5678 1234 68AC 4444
 - Penambahan 0003 dan 0004 = 0007
hasil kurang -1. tdk ada carry maka FFFF.
 - Contoh 2 : penambahan jumlah 5678 dan 1234 adalah 68AC dan hasil kurang adalah 4444

Contoh Perintah Debug

- I (Input) : Membaca byte data dari I/O komputer
 - Contoh : -I 3FD 60
- M (move) : memindah atau menyalin data disuatu lokasi memori
- N (name) : Untuk memberikan nama file pada program com yang telah dibuat tapi belum tersimpan
- O (output)
- P (Proced) : sama dengan instruksi G dan Tuntuk menjalankan
- Q (Quit) : Keluar dari debug, kembali ke Dos promp

Contoh Perintah Debug

- R (Register) : Menampilkan informasi komposisi register di dalam mikroprosesor
 - Syntax :-R atau RCX <value>
C:\>debug
-r
AX= 0 0 0 0 BX= 0 0 0 0 CX= 0 0 0 0
DX= 0 0 0 0 SP= F F E E BP= 0 0 0 0
SI= 0 0 0 0 DI= 0 0 0 0 DS= 137C
ES= 137C SS= 137C CS= 137C
IP= 0100N V UP E I PL NZ NA PO NC
137C:0100 0000 ADD [BX+SI],AL
DS:0000=CD
– RAX = digunakan untuk mengubah isi register AX

-rax

AX 0000

:1234

-t

AX=1234 BX=0000 CX=0000

DX=0000 SP=FFEE BP=0000 SI=0000

DI=0000 DS=137C ES=137C

SS=137C CS=137C IP=0102NV UP

EI PL NZ AC PO CY 137C:0102 0000 ADD

Contoh Perintah Debug

- **RBX : mengubah nilai register BX**
- **T (Trace) : untuk menjalankan baris program rakitan yg sedang di debug.**
 - Trace Tidak sama dengan instruksi
G. Trace menjalankan 1 instruksi,
Go menjalankan semua kode yang di blok pada satu waktu
- **U (unassemble) : Melihat program yang sedang aktif.**
- **W (write) : menyimpan data ke dalam disk.**
- **? : Menampilkan layar help pada versi debug.**

Contoh Perintah Debug

- *MOV: adalah Instruksi untuk memasukkan, memindahkan, mengupdate isi register tertentu.*

Penulisan instruksi:

– **MOV [operand A],[operand B]**

–

D

engan catatan operand A adalah register, variabel atau lokasi memori dan isi operand B berupa register, variabel, lokasi memori ataupun bilangan.

- Operand B merupakan bilangan asal yang akan diisikan ke operand A, dengan kata lain **operand A** merupakan **tujuan pengisian** atau **penduplikatan** dari **operand B**.

– **Contoh:**

– **MOV AH, AL**

- Operand A dari instruksi diatas adalah register AH

- Operand B dari instruksi diatas adalah register AL
- Hal yang dilakukan dari instruksi diatas adalah menduplikatkan isi register AL ke register AH

Contoh Perintah Debug

- *Int (Interrupt)*
- Instruksi Int (Interrupt) bekerja dengan cara yang sama dengan instruksi subroutine.
- Subroutine pada Int (interrupt) terdiridari 2 jenis, yaitu:
- Bios Interrupt yakni Int yang disediakan oleh BIOS meliputi Int 0 hingga Int 1F hexa.
- DOS Interrupt yaitu Int yang disediakan oleh DOS (Disk Operating System). Interrupt yang tergolong dalam kelompok ini diatas Int 1F hexa. Misal: Interrupt 20 hexa, Interrupt 21 hexa dll.

Contoh Perintah Debug

- *Int 20*
- Int 20h adalah satu dari DOS Interrupt. Fungsi Int 20h adalah menghentikan proses komputer terhadap suatu program COM.
- *Int 21h Service 02*
- Int 21h adalah salah satu dari DOS Interrupt. Int 21h mempunyai multi fungsi, maka fungsinya dibagi-bagi. Untuk mengaktifkan bagian itu, diikuti nomor bagiannya yang disebut *Service Number*.
- Untuk mengeksekusi fungsi Int 21h service 02 harus mengikuti ketentuan berikut:
- A. Register AH, harus diisi service number dari Int 21h yang pastinya (02h)
- B. Register DL, harus berisi bilangan hexa dari karakter ASCII yang akan dicetak

Contoh Perintah

- *Prosedur Percobaan*
- Jalankan “DEBUG” pada command prompt
- Ketik A 100 ; Alamat
disegment 100
- MOV AX, 0090 ; masukan data di AX
MOV DS, AX ; Masukkan data DS, dari dataAX
MOV SI, 0120 ; Masukkan data SI
MOV DI, 0130 ; Masukkan data DI
MOV AX, SI ; Masukkan data AX, dari dataSI
MOV DI, AX ; Masukkan data DI, dari dataAX
INT 20H ; End

Contoh Perintah Debug

Menampilkan huruf "A"

C:\>DEBUG

```
-A100 ; Alamat di segment100
158E:0100 MOV AH,02 ;Menyalin Register Ah (8 bit)
158E:0102 MOV DL,41 ;Kode heksa "A"
158E:0104 INT 21 ; Cetak Tampilan(0102)
158E:0106 INT 20 ;End
158E:0108 ;enter
• -G ; Jalankan
```

Contoh Perintah Debug

Menampilkan huruf “AB”

C:\>DEBUG

-A100	; Alamat di segment 100
158E:0100 MOV AH,02	; Menyalin Register Ah (8 bit)
158E:0102 MOV DL,41	; Kode heksa “A”
158E:0104 INT 21	; Cetak Tampilan (0102)
158E:0106 MOV DL,42	; Kode heksa “B”
158E:0108 INT 20	; End

- -G ; Jalankan

Contoh Perintah Debug

Menampilkan huruf “25 ASCII Mode Text”

```
mov ah, 02
```

```
mov cx, 1a mov
```

```
dl, 41
```

```
int 21
```

```
inc dl
```

```
Loop...(alamat register pada int 21)
```

```
Int 20
```

Contoh Perintah Debug

Menampilkan huruf “25 ASCII Mode Grafik”

```
mov ah, 9           ; service 9 int 10 modus grafik
mov al, 5a          ; kode ASCII Z
mov bh, 00          ; nomor halaman 1 (hitam)
mov bl, 04          ; nomor atribut warna merah
mov cx, 001a        ; perulangan sebanyak 26 kali
int 10
dec al              ; pengurangan pada register al
Loop ...(int 10)
Int 20
```

Contoh Perintah Debug

Latihan

1. Buat output untuk menampilkan ABC, AAA, BBB, CCC, ABA, BAC, CCA
2. Amati proses yang terjadi
3. Gunakan Notepad atau wordprocessor dalam editor program
4. Copy dan Paste hasil yang dianggap selesai

Contoh Perintah Debug

Mencetak Baris Berbeda

-A100

158E:0100 MOV AH,02

158E:0102 MOV DL,41

158E:0104 INT 21

158E:0106 MOV DL,0A ;(NOL A) ; Cetak, kursor pindah dibaris
selanjutnya

158E:0108 INT 21

158E:010A MOV DL,42

158E:010C INT 21

158E:010E INT 20

158E:0110

-G

Contoh Perintah Debug

Mencetak Baris Berbeda

-A100

158E:0100 MOV AH,02

158E:0102 MOV DL,41

158E:0104 INT 21

158E:0106 MOV DL,0A ; (NOL A) Cetak, kursor pindah dibaris
selanjutnya

158E:0108 INT 21

158E:010A MOV DL,0D ;(NOL D) Cetak, kursor pindah dibaris
ke awal berikut

158E:010C INT 21

158E:010E MOV DL,42

158E:0110 INT 21

158E:0112 INT 20

158E:0114

-G

Contoh Perintah Debug

Latihan

1. *Buat output untuk menampilkan ABCDEFG*

A
B
C
D
E
F
G

Output-1

A
B
C
D
E
F
G

Output-2

Contoh Perintah Debug

ADD, SUB, INC, dan DEC

- Instruksi **ADD** dan **SUB** digunakan untuk menjumlahkan dan mengurangi isi dua buah register dan suatu lokasi memori, atau menjumlahkan dan mengurangi suatu bilangan ke/dari register atau lokasi memori. Sintaksnya sbb:
- **ADD** tujuan, sumber
- **SUB** tujuan, sumber
- Instruksi **INC** (*increment*) digunakan untuk menambahkan 1 ke register atau lokasi memori, sedangkan **DEC** (*decrement*) digunakan untuk mengurangi 1 isi register atau lokasi memori. Sintaksnya :
- **INC** tujuan

- **DEC tujuan**

Contoh Perintah Debug

MENCETAK BEBERAPA KARAKTER

Untuk mencetak beberapa karakter, bisa anda gunakan proses looping. Sebagai contoh dari penggunaan looping ini bisa dilihat pada program di bawah ini :

A100

0C21:0100 MOV AH,02

0C21:0102 MOV CX,A ;

Banyaknya pengulangan

0C21:0105 MOV DL,41

0C21:0107 INT 21

0C21:0109 INC DL ; Tambah DL dengan 1, **INC DL**
menambah register DL dengan 1

0C21:010B LOOP 107 ; Ulang dan pindah

0C21:010D INT 20

0C21:010F

-G

Contoh Perintah Debug

MENCETAK BEBERAPA KARAKTER

Untuk mencetak beberapa karakter, bisa anda gunakan proses looping. Sebagai contoh dari penggunaan looping ini bisa dilihat pada program di bawah ini :

A100

0C21:0100 MOV AH,02

0C21:0102 MOV CX,F ;

Banyaknya pengulangan (hexa)

0C21:0105 MOV DL,41

0C21:0107 INT 21

0C21:0109 INC DL ; Tambah DL dengan 1, **INC DL**
menambah register DL dengan 1

0C21:010B LOOP 107

; Ulang dan pindah

0C21:010D INT 20

0C21:010F

-G

INTRUKSI STACK

Stack

- Stack merupakan bagian memori yang digunakan untuk menyimpan nilai dari suatu register secara sementara. Operasi stack dinamakan juga LIFO (Last In First Out).
- Bila kita terjemahkan secara bebas, stack artinya adalah 'tumpukan'. Stack adalah bagian memory yang digunakan untuk menyimpan nilai dari suatu register untuk sementara
- Operasi- operasi pada assembler yang langsung menggunakan stack misalnya pada instruksi **PUSH, POP**, PUSF dan POPF.

Contoh Perintah Debug

Pengoperasian Stack :

- *PUSH*
 - Untuk menambahkan sebuah *word* yang baru pada stack kita *PUSH* ke stack. Sintaksnya adalah:
 - **PUSH** *sumber*
 - Eksekusi PUSH menyebabkan terjadinya hal berikut:
 - SP (*stack pointer*) dikurangi 2
 - Salinan isi *sumber* disalin ke alamat yang ditetapkan oleh SS:SP. *Sumber* tidak berubah
- POP
 - Untuk mengambil (remove) item pada puncak stack, kita lakukan **POP**. Sintaksnya adalah:
 - **POP** *tujuan*

- Eksekusi POP menyebabkan terjadinya hal berikut:
- Isi dari SS:SP (puncak stack) disalin ke ***tujuan***
- SP ditambah 2.

Contoh Perintah Debug

- Dengan instruksi "**PUSH**", kita **menyimpan nilai register DX** pada stack, kemudian pada instruksi "**POP**" kita **mangambil keluar nilai yang disimpan tersebut dari stack**. Dari program ini dapat dilihat bagaimana stack menggantikan variabel. yang digunakan untuk menyimpan nilai pada register DX. Untuk mengambil (remove) item pada puncak stack, kita lakukan **POP**

BAB III

Mikrokontroler

Mata Kuliah : Mikoprosessor

Capaian Pembelajaran :

Mahasiswa Mampu menerapkan dan mengembangkan teknik antarmuka, dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi

Kemampuan Akhir yang diharapkan :

Mahasiswa memahami rangkaian sifatnya.

Antarmuka danPeripheral dalam SistemKomputer

Indikator :

- d) .Mahasiswa akan mampu menjelaskan perbedaan antarmuka serial dan parallel
- e) Mahasiswa akan mampu menjelaskan perbedaan mikroprosesor, mikrokontroller dan mikrokomputer

- f) Mahasiswa akan mampu menjelaskan bus yang terdapat dalam mikrokomputer
- g) Mahasiswa akan mampu mengidentifikasi peripheral yang ada di satu sistem mikrokomputer dan antarmukanya

3 Mikrokontroler

3.1. Arsitektur mikro kontroler

Apa isi Mikrokontroler ?

Mikrokontroler tersusun dari beberapa komponen, antara lain :

- ✱ CPU
 - ✱ ROM
 - ✱ RAM
 - ✱ Timer/Counter
 - ✱ Unit I/O (Serial dan Paralel)
-
- Kegunaan dari mikrokontroler pada umumnya banyak digunakan untuk aplikasi sistem kendali atau monitoring, misalnya sebagai alat kontrol penampil tulisan, sistem pengukuran jarak jauh (telemetry), berbagai mainan anak-anak dan sistem elektronika lainnya
 - Mikrokontroler 89C51 merupakan anggota dari keluarga MCS-51 yang diproduksi oleh Intel Corp.
 - Keluarga MCS-51 terdiri atas 4 versi yaitu 8031, 8051, 8751, dan 8951.
 - Tipe 8031 adalah versi tanpa EPROM,

- Tipe 8051 adalah versi dengan 4 Kbyte ROM
- Tipe 8751 adalah versi 4 Kbyte EPROM
- Tipe 89C51 adalah versi dengan EEPROM
- Kode C menyatakan CMOS. Chip 89C51 disebut juga *flash microcontroller*.

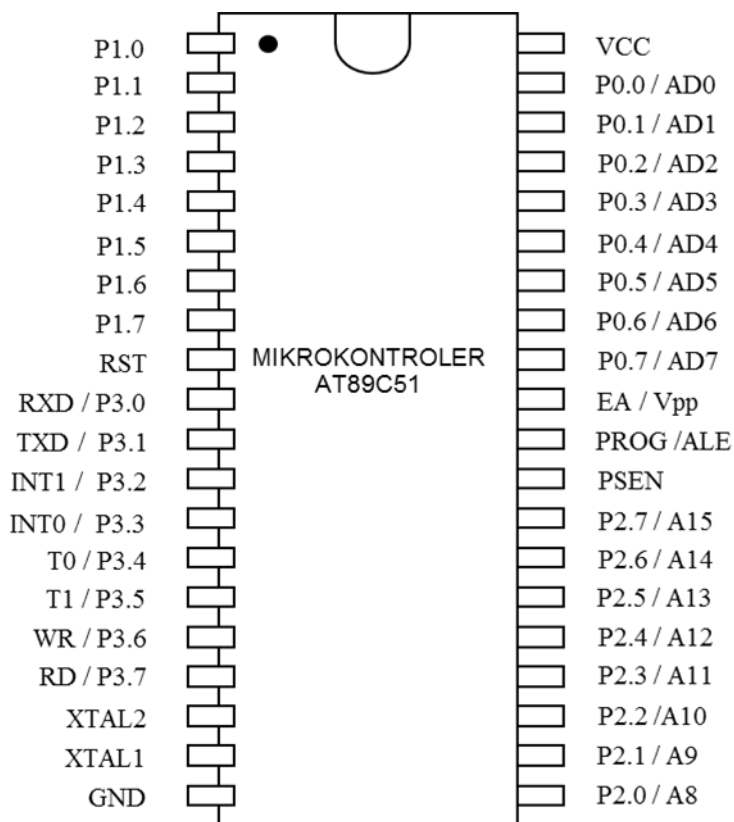
Spesifikasi Mikrokontroler 89C51

- ☀ Mempunyai 40 kaki atau pin
- ☀ 4 Kbyte EEPROM (Electrical Eraseable Programmable Read Only Memory) untuk memori program
- ☀ 128 byte internal RAM untuk memori data
- ☀ 128 byte Register khusus (SFR)
- ☀ 4 buah port I/O yang masing-masing terdiri dari 8 bit, sifatnya dua arah, setiap bit dapat dialamatasi
- ☀ 2 buah timer/counter 16 bit
- ☀ 2 buah Interupsi
- ☀ Sebuah port serial full duplex, yaitu port 3
- ☀ Bekerja pada frekuensi 0-12 MHz, dengan osilator internal

- ☀ Mempunyai fasilitas penguncian program untuk menghindari terjadinya pembajakan program

Bagaimana Susunan Kaki-Kaki Mikrokontroler 89C51 ?

- Kaki-kaki 89C51 terdiri atas 40 kaki dengan susunan seperti gambar berikut :



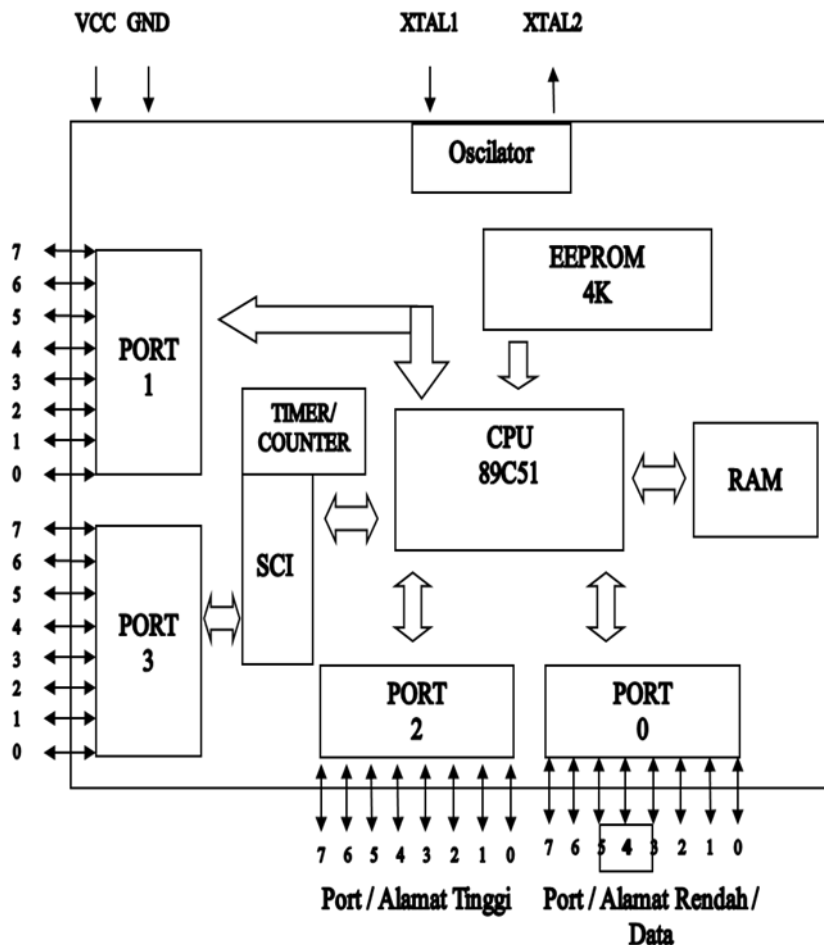
Apa fungsi masing-masing kaki ?

- RST : kaki ini digunakan untuk mereset mikrokontroler pada transisi rendah ke tinggi

- ALE/PROG : kaki ini digunakan mikrokontroler untuk meng-enable alamat byte rendah A0-A7 bila digunakan external EPROM, tetapi bila digunakan internal EPROM, maka kaki ini berfungsi untuk menerima pulsa program selama proses pemrograman EPROM
- PSEN : (*Program Store Enable*) adalah sinyal keluaran yang mengaktifkan External EPROM
- EA/Vpp : Pada kondisi tinggi, mikrokontroler menjalankan perintah dari internal EPROM. Pada kondisi rendah, mikrokontroler menjalankan perintah dari external EPROM. Pada saat pemrograman EPROM dalam, kaki ini menerima tegangan 21 Volt untuk pemrograman EPROM
- XTAL1 : Masukan ke penguat osilator atau sumber masukan osilator luar
- XTAL2 : Output penguat osilator
- P0 : Dapat berfungsi sebagai port atau sebagai bus alamat dan bus data
- P2 : Dapat berfungsi sebagai port atau sebagai bus alamat (byte atas)

- P3 : Dapat berfungsi sebagai port atau fungsi lain untuk sinyal kendali (misal dalam komunikasi serial, masukan interupsi, masukan untuk pencacah)
- AD0 – AD7 : Berfungsi sebagai bus alamat dan data (A = Alamat, D = Data)
- A8 – A15 : Berfungsi sebagai bus alamat (bit alamat 8 s/d 15)
- Mikrokontroler 89C51 sudah dilengkapi dengan osilator di dalam chip, dengan frekuensi kerja antara 1,2 MHz – 12 MHz.
- Dapat juga digunakan kristal atau pembangkit Clock luar (external)

Bagaimana Arsitektur 89C51 ?



Ada Berapa Port Input dan Output ?

- Dalam 89C51 terdapat 32 jalur port yang dikelompokkan dalam 4 buah port yang masing-masing terdiri dari 8 bit.

- Masing-masing bit dapat diprogram secara individual dan bebas sebagai input maupun output dan dapat dikonfigurasi secara dinamis lewat perangkat lunak.

Bagaimana Konfigurasi Tiap Port ?

- ✱ PORT 0 tidak mempunyai Pull-Up internal. Pull-Up FET yang ada hanya digunakan saat pengaksesan memori eksternal
- ✱ PORT 1, 2, dan 3 mempunyai Pull-Up internal. Pada penggunaan P1, P2, P3 sebagai input, pin-pinnya di Pull-Up tinggi oleh Pull-Up internal, dan dapat di Pull-Up rendah oleh sumber eksternal. Dengan mengisi 1 ke latch, maka port tersebut akan berfungsi sebagai input.

Apakah Fungsi Timer ?

- Chip ini mempunyai 2 buah Timer 16 bit.
- Timer dapat dijalankan dalam 4 mode.
- Timer dapat ditulis atau dibaca setiap saat, termasuk saat timer sedang jalan.
- Clock dapat diambilkan dari dalam (dengan instruksi) atau dari luar. Bila Timer overflow akan terjadi interupsi.

Apakah Fungsi Port Serial ?

- Port serial merupakan peripheral yang sangat kompleks, dapat mengirim data secara serial baik dalam mode sinkron atau tak sinkron.
- Cara Kerja port serial ini dikendalikan oleh register SCON (dalam alamat 098h)

Apakah Interupsi itu ?

- ☀ Interupsi merupakan bagian yang penting dalam mikrokontroler. Fasilitas ini dapat digunakan untuk meminta pelayanan dari mikrokontroler.
- ☀ Interupsi bisa diterima atau ditolak tergantung dari setting dalam pemrogramannya. Mikrokontroler ini mempunyai 8 fasilitas interupsi, yang masing-masing mempunyai vektor yang terpisah.

3.2. Rangkaian clock dan reset.

RESET CIRCUIT OF 8051 CONTROLLER

RESET adalah input Tinggi yang aktif Ketika RESET diatur ke Tinggi, 8051 kembali ke status daya. 8051 diatur ulang dengan menahan RST tinggi untuk setidaknya dua siklus mesin dan kemudian mengembalikannya rendah.

Ada dua metode rangkaian reset:

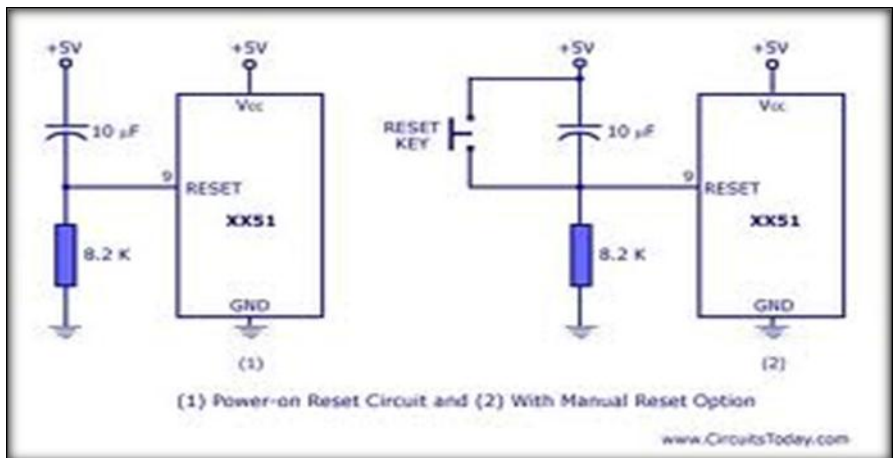
1. Power Di Reset.

Awalnya pengisian kapasitor membuat RST Tinggi

Ketika kapasitor mengisi penuh, ia memblokir DC.

2. Reset Manual

menutup sakelar sebentar akan membuat RST Tinggi.



Setelah reset, penghitung program dimuat dengan 0000H tetapi isi dari chip on-chip tidak terpengaruh.

Register	Content
Program counter	0000h
Accumulator	00h
B register	00h
PSW	00h
SP	07h
DPTR	0000h
All ports	FFh

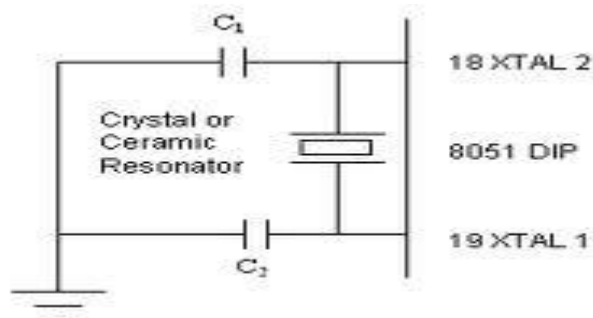
Catatan: konten RAM pada chip tidak terpengaruh oleh Reset.

RANGKAIAN OSCILLATOR 8051 CONTROLLER

8051 menggunakan kristal untuk hal itu untuk menyinkronkan operasi itu. Secara efektif, 8051 beroperasi menggunakan apa yang disebut "siklus mesin."

Satu siklus mesin adalah jumlah waktu minimum di mana instruksi 8051 tunggal dapat dieksekusi. walaupun banyak instruksi mengambil beberapa siklus.

8051 memiliki osilator on-chip. Dibutuhkan kristal eksternal yang menentukan frekuensi operasi 8051.

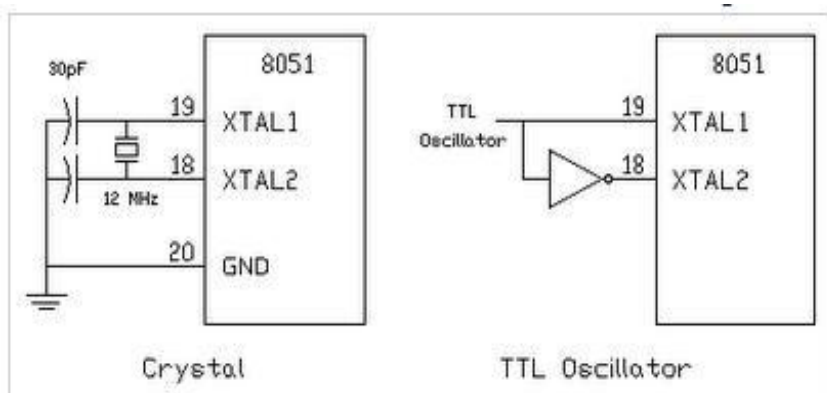


ini bisa dicapai dengan dua cara.

Kristal terhubung ke pin 18 dan 19 dengan kapasitor penstabil.

Kristal 12 MHz (11.059MHz) sering digunakan dan kapasitansinya berkisar dari 20pF hingga 40pF.

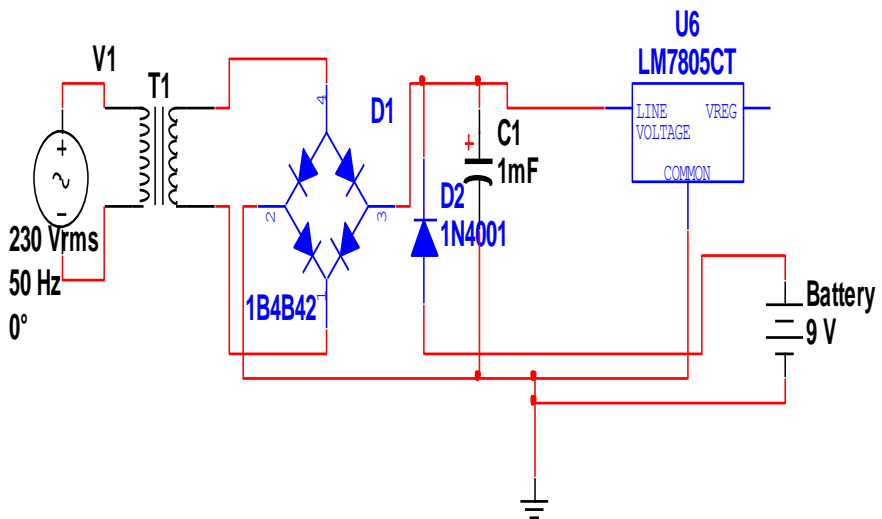
Osilator juga dapat menjadi sumber jam TTL yang terhubung dengan gerbang NOT seperti yang ditunjukkan.



3.3. Power suplay

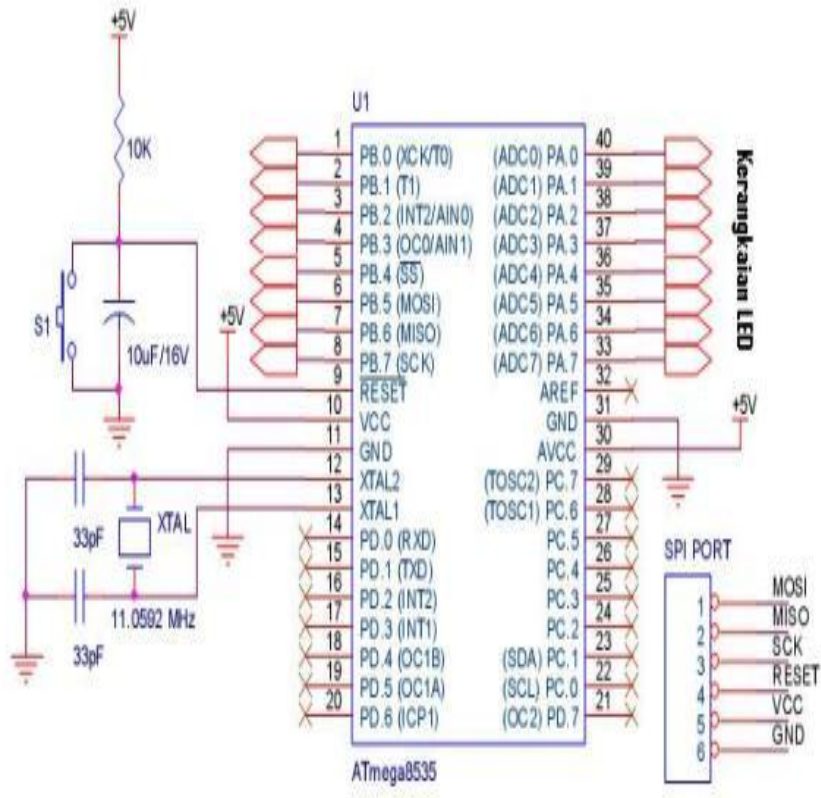
Unit catu daya

Catu daya (kadang-kadang dikenal sebagai unit catu daya atau PSU) adalah perangkat atau sistem yang memasok energi listrik atau jenis lain ke beban keluaran atau kelompok beban. Istilah ini paling sering diterapkan pada pasokan energi listrik, lebih jarang untuk yang mekanis, dan jarang untuk orang lain.



Bagian daya mengambil tegangan listrik AC pada inputnya dan memberikan output DC 5V yang stabil.

Contoh Project minimum sistem



BAB IV

Arduino Mikrokomputer

Mata Kuliah : Mikroprosesor

Capaian Pembelajaran :

Mahasiswa Mampu menerapkan dan mengembangkan teknik antarmuka dengan arduino , dalam sistem mikro komputer sesuai dengan kebutuhan desain yang harus dipenuhi

Kemampuan Akhir yang Diharapkan :

Memahami Arduino dan Dasa Antar

Indikator :

1. Mahasiswa akan mampu menjelaskan antarmuka yang disediakan oleh mikrokontroler Arduino
2. Mahasiswa akan mampu menjelaskan antarmuka yang disediakan oleh mikrokontroler Arduino
3. Mahasiswa akan mampu mengaplikasikan rangkaian arduino ke system kontrol

4. Mahasiswa akan mampu mengaplikasikan rangkaian input / output ke sistem mikrokontroler Arduino sesuai dengan kebutuhan dan konstrain kecepatan sistem kontrol
5. Mahasiswa akan mampu mengaplikasikan rangkaian kontrol arduino ke system mikrokontroler dengan tepat

4. ARDUINO

4.1. Pengertian Arduino

Arti dari Arduino disini adalah sebuah rangkaian elektronik yang memiliki suatu komponen utama chip mikrokontroler. Mikrokontroler yang disebutkan disini adalah sebagai sebuah chip atau IC yang dapat diprogram melalui komputer. Arduino juga adalah kit elektronik atau papan rangkaian elektronik open source yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel. Tujuan dari adanya mikrokontroler untuk membaca input dengan memprosesnya hingga menimbulkan Output.



Dari adanya Mikrokontroler, ini dapat ditemukan pada berbagai alat seperti handphone, MP3 player, DVD, televisi, AC

dan masih banyak lagi. Selain itu, Mikrokontroler juga berguna sebagai pengendali robot baik mainan atau dari industri. Disini juga menyebutkan bahwa komponen utama Arduino adalah mikrokontroler.

Selain dari pengertian diatas, ada juga pengertian lain yang menyatakan bahwa Arduino merupakan pengendali tunggal yang disebut sebagai mikro board dengan sifat terbuka atau open source. Alat ini dirancang dengan tujuan untuk memudahkan penggunaan berbagai macam alat elektronik yang salah-satunya adalah komputer. Dalam penggunaannya, arduino memiliki prosesor Atmel AVR serta program bahasa tersendiri. Dari penjelasan yang telah disebutkan di atas, Arduino juga menyimpan sejarah yang akan diungkapkan dibawah ini.

Sekitar tahun 2005, penciptaan Arduino berasal dari teori oleh Hernando Barragan di Italia. Dari adanya teori tersebut, kemudian ini dikembangkan oleh Massimo Banzi dan David Cuartielles yang menamainya sebagai Arduin of Ivrea. Kemudian, alat tersebut berganti nama yang diambil dari bahasa Italia menjadi Arduino.

Berdasarkan dari sifat yang dimilikinya, perkembangan Arduino sangat cepat. Dari perkembangan tersebut, banyak penciptaan sejenis Arduino seperti DFRduino atau Freeduino, CipaDuino, MurmerDuino dan AViShaDuino.

Seiring waktu, ada berbagai macam jenis penciptaan Arduino dari yang mudah dicari dan yang paling banyak digunakan yang salah-satunya seperti Arduino Uno. Berdasarkan penggunaannya, telah tercatat bahwa tahun 2011 telah ada ratusan ribu Arduino digunakan. Dari lahirnya alat tersebut, banyak perusahaan besar yang menggunakan alat tersebut seperti Google, NASA dan juga Large Hadron ColliderMurah – Papan (perangkat keras) Arduino biasanya dijual relatif murah (antara 125 ribu hingga 400 ribuan rupiah saja) dibandingkan dengan platform mikrokontroler pro lainnya. Jika ingin lebih murah lagi, tentu bisa dibuat sendiri dan itu sangat mungkin sekali karena semua sumber daya untuk membuat sendiri Arduino tersedia lengkap di website Arduino bahkan di website-website komunitas Arduino lainnya. Tidak hanya cocok untuk Windows, namun juga cocok bekerja di Linux.

Sederhana dan mudah pemrogramannya – Perlu diketahui bahwa lingkungan pemrograman di Arduino mudah

digunakan untuk pemula, dan cukup fleksibel bagi mereka yang sudah tingkat lanjut. Untuk guru/dosen, Arduino berbasis pada lingkungan pemrograman Processing, sehingga jika mahasiswa atau murid-murid terbiasa menggunakan Processing tentu saja akan mudah menggunakan Arduino.

Perangkat lunaknya Open Source – Perangkat lunak Arduino IDE dipublikasikan sebagai Open Source, tersedia bagi para pemrogram berpengalaman untuk pengembangan lebih lanjut. Bahasanya bisa dikembangkan lebih lanjut melalui pustaka-pustaka C++ yang berbasis pada Bahasa C untuk AVR.

Perangkat kerasnya Open Source – Perangkat keras Arduino berbasis mikrokontroler ATMEGA8, ATMEGA168, ATMEGA328 dan ATMEGA1280 (yang terbaru ATMEGA2560). Dengan demikian siapa saja bisa membuatnya (dan kemudian bisa menjualnya) perangkat keras Arduino ini, apalagi bootloader tersedia langsung dari perangkat lunak Arduino IDE-nya. Bisa juga menggunakan breadboard untuk membuat perangkat Arduino beserta periferal-periferal lain yang dibutuhkan.

4.2. Kelebihan Arduino

Tidak perlu perangkat chip programmer karena didalamnya sudah ada bootloader yang akan menangani upload program dari komputer. Sudah memiliki sarana komunikasi USB, Sehingga pengguna laptop yang tidak memiliki port serial/RS323 bisa menggunakannya. Memiliki modul siap pakai (Shield) yang bisa ditancapkan pada board arduino. Contohnya shield GPS, Ethernet,dll.

Jenis - Jenis Arduino

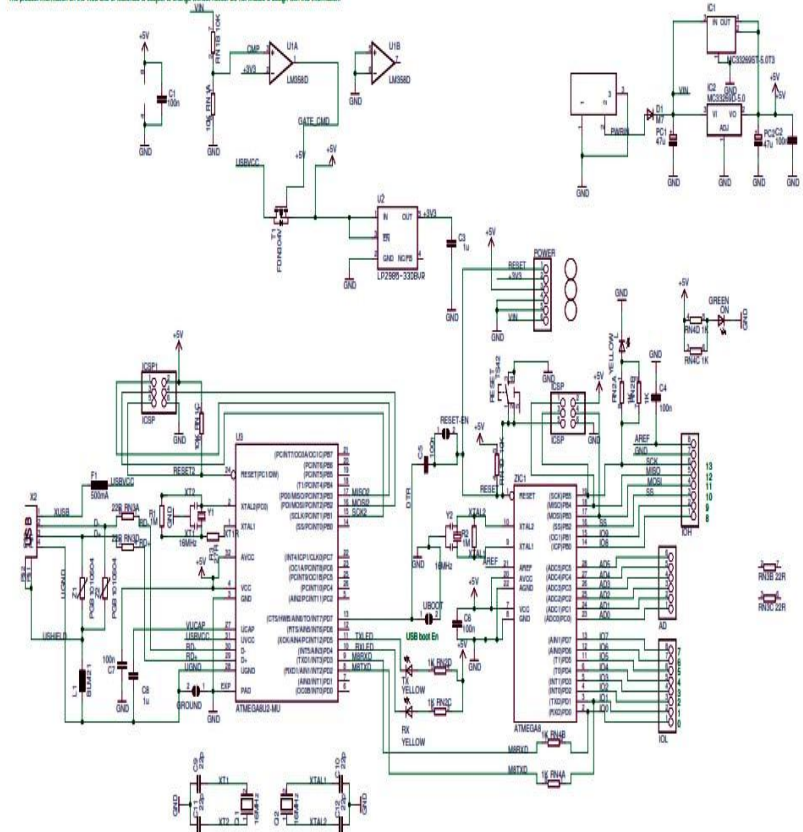
- | | |
|---------------------|---------------------|
| 1. Arduino Uno | 3. Arduino Mega |
| 2. Arduino Due | 4. Arduino Leonardo |
| 5. Arduino Nano | 9. Arduino Esplora |
| 6. Arduino Mini | 10. Arduino Robot |
| 7. Arduino Micro | 11. Arduino Fio |
| 8. Arduino Ethernet | 12. Arduino |
| 13. LilyPad | |

Digram rangkaian Arduino Uno

Arduino™ UNO Reference Design

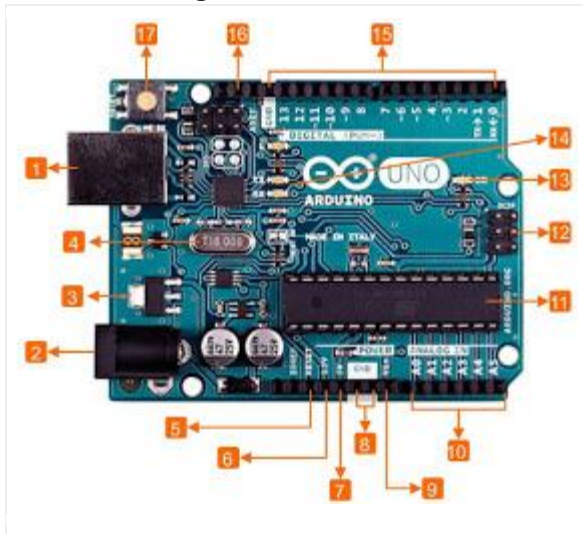
Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Arduino may make changes to specifications and product descriptions at any time without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Arduino reserves the right to change specifications and product descriptions at any time without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined".



4.3. Bagian –Bagian Arduino

Arduino (Uno R3) pada umumnya menggunakan ATMEGA328. Adapun bagian-bagian dari papan Arduino UNO ini adalah sebagai berikut :



1) Power USB

USB Soket/Power USB digunakan untuk memberikan catu daya ke Papan Arduino menggunakan kabel USB dari komputer. Selain menjadi port catu daya, USB juga memiliki berfungsi untuk:

- i. Memuat program dari komputer ke dalam board Arduino.

ii. Komunikasi serial antara papan Arduino dan komputer begitu juga sebaliknya.

2) Power (*Barrel Jack*)

Arduino ini juga diberi colokan untuk catu daya.

Tegangan maksimal yang dapat diberikan kepada

Arduino maksimal 12V dan maksimal arusnya adalah 2A.

3) Voltage Regulator

Menurunkan tegangan yang diberikan ke papan Arduino

lalu menstabilkan tegangan DC yang digunakan oleh

prosesor dan komponen - komponen lain.

4) Crystal Oscillator

Kristal (*quartz crystal oscillator*), jika mikro kontroler

dianggap sebagai sebuah otak, maka kristal adalah

jantung – nya karena komponen ini menghasilkan

detak-detak yang dikirim kepada mikrokontroler agar

melakukan sebuah operasi untuk setiap detak-nya.

Kristal ini dipilih yang berdetak 16 juta kali per detik

(16MHz).

5) 5, 17 Tombol Reset

Kita dapat mereset papan arduino, misalnya memulai program dari awal. Perhatikan bahwa tombol reset ini bukan untuk menghapus program atau mengosongkan mikrokontroler.

- 6) 3.3V (6) – Supply 3.3 output volt
- 7) 5V (7) – Supply 5 output volt
- 8) GND (8) (Ground)
- 9) Vin (9) – Sumber daya.
- 10) 10 Kaki analog.

Arduino Uno punya enam pin input analog A0 sampai A5. Pin-pin ini dapat membaca tegangan dan sinyal yang dihasilkan oleh sensor analog seperti sensor kelembaban atau temperatur dan mengubahnya menjadi nilai digital yang dapat dibaca oleh mikroprosesor.

11) Main microcontroller

Setiap papan Arduino memiliki Mikrokontroler (11). Kita dapat menganggapnya sebagai otak dari papan Arduino. IC (integrated circuit) utama pada Arduino sedikit berbeda antara papan arduino yang satu dengan yang lainnya. Mikrokontroler yang sering digunakan adalah ATMEGA. Kita harus mengetahui IC apa yang dimiliki oleh suatu papan Arduino sebelum memulai memprogram arduino melalui Arduino IDE. Informasi tentang IC

terdapat pada bagian atas IC. Untuk mengetahui konstruksi detail dari suatu IC, kita dapat melihat lembar data dari IC yang bersangkutan.

12) 12 ICSP pin

Kebanyakan, ICSP (12) adalah AVR, suatu programming

Ringkasan

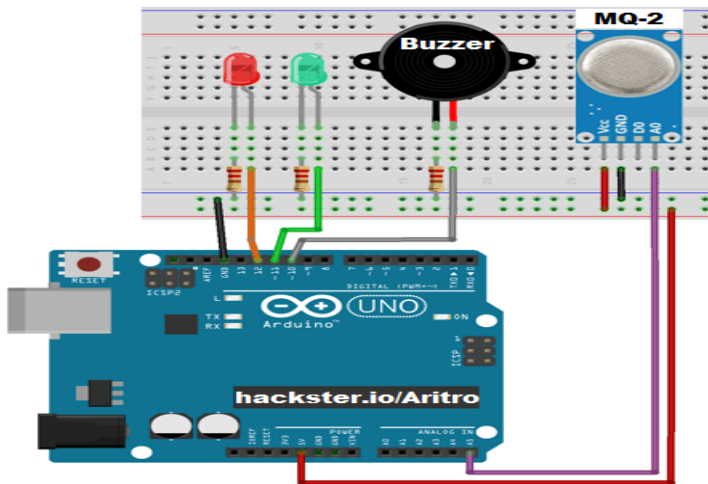
Mikrokontroler	ATmega328
Tegangan pengoperasian	5V
Tegangan input yang disarankan	7-12V
Batas tegangan input	6-20V
Jumlah pin I/O digital	14 (6 di antaranya menyediakan keluaran PWM)
Jumlah pin input analog	6
Arus DC tiap pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
Memori Flash	32 KB (ATmega328), sekitar 0.5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Bab V

Implentasi dari Arduino

5.1. ALAT PENDETEKSI KEBOCORAN GAS BERBASIS ARDUINO UNO

1. Bahan
 - a) Arduino UNO
 - b) Breadboard
 - c) Sensor MQ-2
 - d) Resistor 220 Ohm
 - e) Male to Male Jumper
 - f) 5 mm LED: Red
 - g) 5 mm LED: Green
 - h) Buzzer
2. wiring



3. Coding Arduino

/******

All the resources for this project:
<https://www.hackster.io/Aritro>

*****/

```
intredLed=12;
intgreenLed=11;
intbuzzer=10;
intsmokeA0=A5;
// Your threshold value
intsensorThres=100;
```

```
voidsetup() {
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(smokeA0, INPUT);
  Serial.begin(9600);
}
```

```
voidloop() {
  intanalogSensor=analogRead(smokeA0);
```

```
  Serial.print("Pin A0: ");
  Serial.println(analogSensor);
  // Checks if it has reached the threshold value
  if (analogSensor>sensorThres)
  {
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    tone(buzzer, 1000, 200);
```

```
    }  
    else  
    {  
        digitalWrite(redLed, LOW);  
        digitalWrite(greenLed, HIGH);  
        noTone(buzzer);  
    }  
    delay(100);  
}
```

4. Cara Kerja Alat

Ketika sensor MQ2 mendeteksi adanya gas, maka Buzzer akan mengeluarkan suara dan LED warna merah akan menyala.

Dalam keadaan normal, LED warna hijau akan terus menyala, menandakan bahwa keadaan ruangan aman dari kebocoran gas.

5.2. Sensor intensitas cahaya

Komponen :

1. 1x Arduino
2. 1x Breadboard
3. 1x Sensor cahaya LDR
4. 1x Resistor 10k
5. 7x Kabel jumper

Wiring / Hubungan komponen :

- Hubungkan **5V** dan **GND** dari **Arduino** ke **Breadboard**.
- Hubungkan **kaki kiri LDR** ke **5V**.
- Hubungkan **kaki kanan LDR** ke pin **A2 Arduino**.
- Hubungkan **kaki kiri resistor** kecelahantara **kaki kanan LDR** dan **GND arduino**.
- Hubungkan **kaki kanan resistor** ke **GND**.
- Untuk pemasangan LED
kamubisalihtpada **Menyalakan LED dan BLINK**.

Coding dari Arduino :

```
1 byte ldr= A2;  
2 byte led= 13;  
3 int nilai;  
4  
5 void setup(){
```

```
6  pinMode(led, OUTPUT);
7  Serial.begin(9600);
8  }
9
10 void loop(){
11  nilai= analogRead(ldr);
12  Serial.print("Nilai LDR: ");
13  Serial.println(nilai);
14
15  if(nilai< 500){
16    digitalWrite(led, HIGH);
17  }
18  else{
19    digitalWrite(led, LOW);
20  }
21
22 }
```

5.3. Mengendalikan LED Menggunakan Perintah Suara

Komponen :

- Arduino uno
- HC – 05 bluetooth
- 4 buah LED
- Kabel jumper
- USB

Pemasangan pada arduino :

4 buah led terhubung dengan pin digital 2 3 4 5

RX dan TX pada hc 05 terhubung dengan pin 10 11

Ground pada hc 05 terhubung dengan led

5v pada hc 05 terhubung dengan pin 5v pada Arduino sebagai sumber

Koding arduino

```
#include <SoftwareSerial.h>
```

```
String value;
```

```
intTxD = 11;
```

```
intRxD = 10;
```

```
SoftwareSerialbluetooth(TxD, RxD);
```

```
void setup(){
```

```
pinMode (2,OUTPUT);
```

```
pinMode (3,OUTPUT);
```

```
pinMode (4,OUTPUT);
```

```
pinMode (5,OUTPUT);
```

```

Serial.begin(9600);
bluetooth.begin(9600);
}

void loop (){
  Serial.println(value);
  if(bluetooth.available())
  {
    value = bluetooth.readString();
    if (value == "kamartidur on"){
      digitalWrite(2,HIGH);
    }
    if (value == "kamartidur off"){
      digitalWrite(2,LOW);
    }
    if(value == "terasrumah on"){
      digitalWrite(3,HIGH);
    }
    if(value == "terasrumah off"){
      digitalWrite(3,LOW);
    }
    if(value == "kamarmandi on"){
      digitalWrite(4,HIGH);
    }
    if(value == "kamarmandi off"){
      digitalWrite(4,LOW);
    }
    if(value == "kipas on"){
      digitalWrite(5,HIGH);
    }
    if(value == "kipas off"){
      digitalWrite(5,LOW);
    }
    if(value == "semua on"){
      digitalWrite(2,HIGH);
    }
  }
}

```

```
digitalWrite(3,HIGH);  
digitalWrite(4,HIGH);  
  
digitalWrite(5,HIGH);  
}  
if(value == "semua off"){  
digitalWrite(2,LOW);  
  
        digitalWrite(3,LOW);  
digitalWrite(4,LOW);  
digitalWrite(5,LOW);  
}  
  
}  
}
```

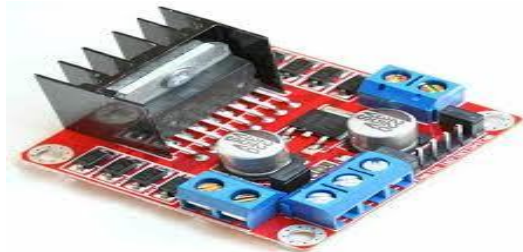
5.4. Robot Penghalang Rintangan

➤ **Komponen**

- Sensor ultrasonic HC – SR04
Berfungsi untuk menghitung jarak dari suatu objek yang ada di depan sensor.



- Driver Motor L298N
Berfungsi untuk mengontrol kecepatan dan arah penggerak motor.



- Kabel Jumper
Berfungsi untuk mengalirkan arus/tegangan dan perintah

program dari pin arduino ke motor driver L298N atau juga bisa ke sensor jarak/ultrasonik



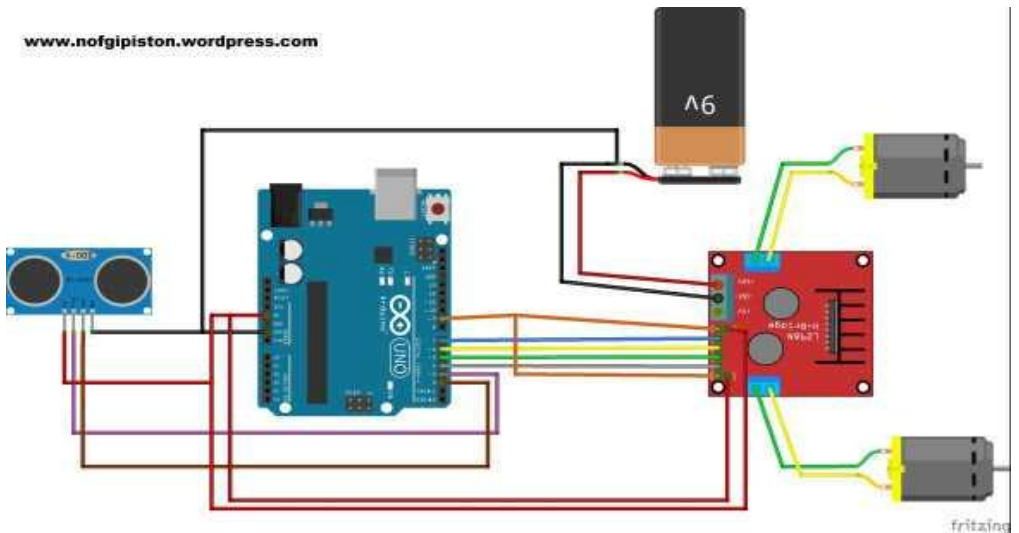
- ModulArduino Uno
Berfungsi sebagai platform open source electronick yang mudah digunakan baik dari sisi hardware maupun software
- Motor dc 6v + gear box
Berfungsi untuk bergerak dari satu



tempat ke tempat yang lain yang diatur oleh arduino dan motor driver L298N

- Power Supply ± 9 volt
Berfungsi untuk men-supply daya

Wiring:



Sistem Kerja Alat :

Robot penghalang rintang ini menggunakan sensor ultrasonik/sensor jarak diumpamakan sebagai mata dari robot ini lalu

ada arduino yang memprogram dari inputan dari ultrasonik tersebut lalu diproses ke driver motor L298N lalu dikeluarkan lewat motor dc 6volt.

wiring Driver L298N dengan Arduino UNO

DRIVER L298N	PIN ARDUINO
ENA-BAWAH	9
ENB-BAWAH	9
ENA-ATAS	5VOLT
ENB-ATAS	5VOLT
PIN IN1	4
PIN IN2	5
PIN IN3	6
PIN IN 4	7

wiring Sensor Ultrasonic:

PIN SENSOR ULTRASONIC	PIN ARDUINO
VCC	+5V
GND	GND

TRIG	3
ECHO	2

Source Code / Sketch:

Harus diisi

```
#include <L298N.h>
```

```
//Untuk pin yang akan menjalankan motor  
kiri dan motor kanan
```

```
const int motor1=7;
```

```
const int motor2=6;
```

```
const int motor3=5;
```

```
const int motor4=4;
```

```
//Untuk pin yang menjalankan sensor ultrasonik
```

```
const int trig=3;
```

```
const int echo=2;
```

```
//untuk variable durasi dan jarak ultrasonik
```

```
int durasi , jarak;
```

```
//Untuk pin pengaturan kecepatan wajib  
menggunakan pin PWM
```

```
const int pinSpeed=9;
```

```
//inisialisasi untuk variable penampung nilai  
kecepatan
```

```

int Speed;
int maju;
void setup() {
    // put your setup code here, to run once:

    Serial.begin(9600);    //      begin      serial
                           communication
    pinMode(motor1,OUTPUT);
    pinMode(motor2,OUTPUT);
    pinMode(motor3,OUTPUT);
    pinMode(motor4,OUTPUT);

    pinMode(trig,OUTPUT);
    pinMode(echo,INPUT);

    pinMode(pinSpeed,OUTPUT);
    pinMode(pinmaju,OUTPUT);

}

void loop() {
    //-----pengaturan kecepatan-----
    //membatasinilai speed
    Speed =(Speed,0,250);
    maju=(maju,0,250);
    //nilai kecepatan dapat diubahnangkene

```

```

Speed=150;
maju=150;
//menuliskan nilai speed pada pin speed
analogWrite(pinSpeed, Speed);
analogWrite(pinmaju, maju);

//-----MENGAKTIFKAN
SENSOR ULTRASONIK BOS----
//mengaktifkan pin trigger
digitalWrite(trig,HIGH);
delayMicroseconds(10);
digitalWrite(trig,LOW);

//mendapat pantulan gelombang ultrasonik
durasi=pulseIn(echo,HIGH);
//Konversi durasi kejarak dalam satuan
centimeter
jarak=((durasi*0.034)/2);

//-----MENGATUR
PERGERAKAN ROBOT-----
//Jika jarak lebih atau sama dengan 20cm
if (jarak>=20)
{

```

```
digitalWrite(motor1,LOW);  
digitalWrite(motor2,HIGH);  
digitalWrite(motor3,LOW);  
digitalWrite(motor4,HIGH);  
}  
else if (jarak<=15)  
{  
digitalWrite(motor1,HIGH);  
digitalWrite(motor2,LOW);  
digitalWrite(motor3,HIGH);  
digitalWrite(motor4,LOW);
```

```
digitalWrite(motor1,HIGH);  
digitalWrite(motor2,LOW);  
digitalWrite(motor3,LOW);  
digitalWrite(motor4,HIGH);
```

```
}  
else if (jarak<=15)  
{  
//jalan  
digitalWrite(motor1,LOW);  
digitalWrite(motor2,HIGH);  
digitalWrite(motor3,LOW);  
digitalWrite(motor4,HIGH);
```

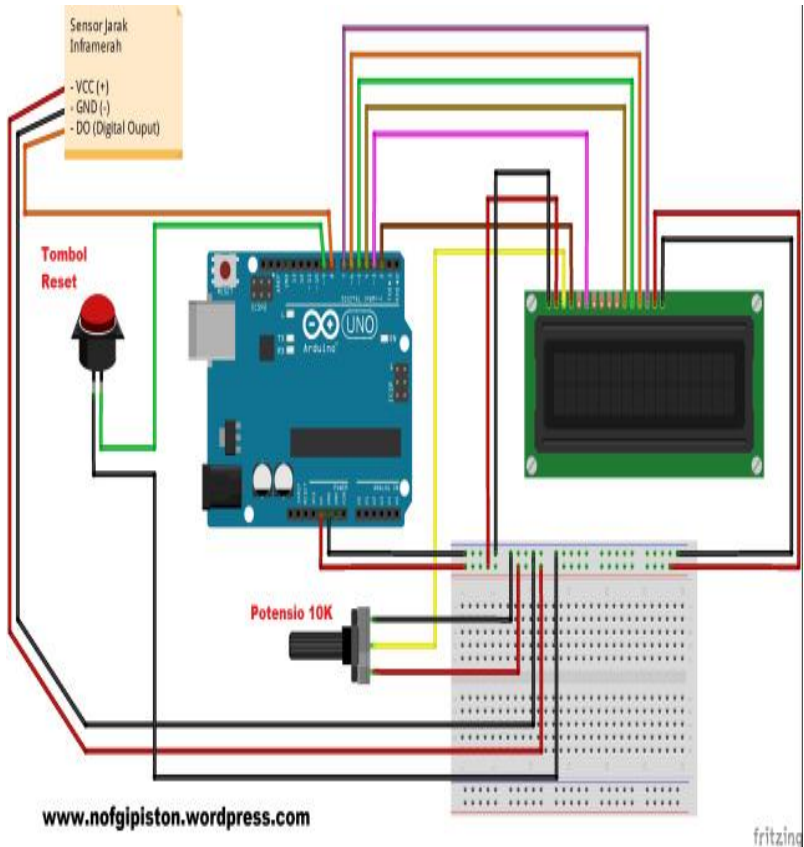
```
digitalWrite(motor1,HIGH);  
digitalWrite(motor2,LOW);  
digitalWrite(motor3,LOW);  
digitalWrite(motor4,HIGH);  
}  
}
```

5.5. Penghitung Barang Berbasis Arduino

➤ **Komponen :**

- Sensor Jarak Inframerah
- Modul LCD 2 X 16
Berfungsi sebagai media tampilan (output) suatu data, baik karakter atau huruf
- I2C LCD
Berfungsi Sebagai Untuk Alur Komunikasi Data Pada LCD
- Push Button
Berfungsi sebagai tombol reset angka counter
- Modul Arduino Uno
- Power Supply \pm 5volt
- Kabel Jumper Arduino

Wiring:



Koneksi Arduino UNO dengan LCD menggunakan modul I2C :

PIN ARDUINO	LCD
5V	VCC
GND	GND
SCL	A5
SDA	A4

Koneksi Sensor Jarak Inframerah :

PIN SENSOR ULTRASONIC	PIN ARDUINO
VCC	+5V
GND	GND
OUT	PIN 11

Koneksi Push Button :

KONEKSI BUZZER	PIN ARDUINO
INPUT	Pin 13
OUTPUT	GND

➤ **Sistem Kerja Alat :**

Sensor penghitung barang ini inputannya menggunakan sensor inframerah. Ketika ada barang yang melintas di depan sensor inframerah barang akan langsung terhitung. Untuk batasan jumlah hitungan dapat diatur diprogram arduno.

Source Code/ Sketch:

```
// memanggil library lcd
#include <LiquidCrystal.h>

// inialisasi pin lcd
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

// inialisasi masing2 pin
const int pinSensor = 8;
const int pinReset = 9;

// inialisasi masing2 variabel
int hitung = 0;
int kondisi1 = 0;
int status1;

//***** ----- program default/setting awal -----
*****//
void setup()
```

```

{
  // inialisasi status pin reset
  pinMode(pinReset, INPUT);
  digitalWrite(pinReset, HIGH);

  pinMode(pinSensor, INPUT);

  // inialisasi jumlah baris-kolom lcd
  lcd.begin(16, 2);

  // tulisan awal pada lcd
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Alathitung");
  lcd.setCursor(0, 1);
  lcd.print("Jumlahng ...");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Ready");
  delay(1000);
}

//***** ----- PROGRAM UTAMA -----
*****//

void loop()
{
  // ----- program penghitung barang ----- //
  / setting maksimal pembacaan jumlah barang

```

```

// ganti jumlah sesuai kebutuhan
hitung = constrain(hitung, 0, 10); // ==> jumlah maks
barang

// status1 adalah hasil pembacaan pin sensor
status1 = digitalRead(pinSensor);

// jika pin sensor bernilai logic HIGH
if (status1 == HIGH)
{
    // hasil hitung tetap
    hitung = hitung;
    kondisi1 = 0;
}

// jika pin sensor bernilai LOW dan kondisi1 bernilai = 0
else if (status1 == LOW && kondisi1 == 0)
{
    // jumlah barang bertambah 1
    hitung += 1;
    // kondisi1 menjadi bernilai = 1
    kondisi1 = 1;
}

// jika pin sensor bernilai LOW dan kondisi bernilai = 1
else if (status1 == LOW && kondisi1 == 1)
{
    // hasil hitung tetap

```

```

hitung = hitung;
// kondisi1 tetap bernilai =1
kondisi1 = 1;
}

// ----- program tombol reset kembali ke = 0 ----- //
// jika tombol reset ditekan
// maka pin reset bernilai logic LOW
if (digitalRead(pinReset) == LOW)
{
// jumlah barang kembali menjadi = 0
hitung = 0;
// menuliskan pada lcd
lcd.clear();
/
lcd.setCursor(0, 0);
lcd.print("RESETf ...");
delay(1000);
}
// ----- program tampilan jumlah barang pada lcd 16x2 ---
--- //
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Jumlahng : ");
lcd.setCursor(0, 1);
lcd.print(hitung);
delay(300); // delay update tulisan pada lcd
}

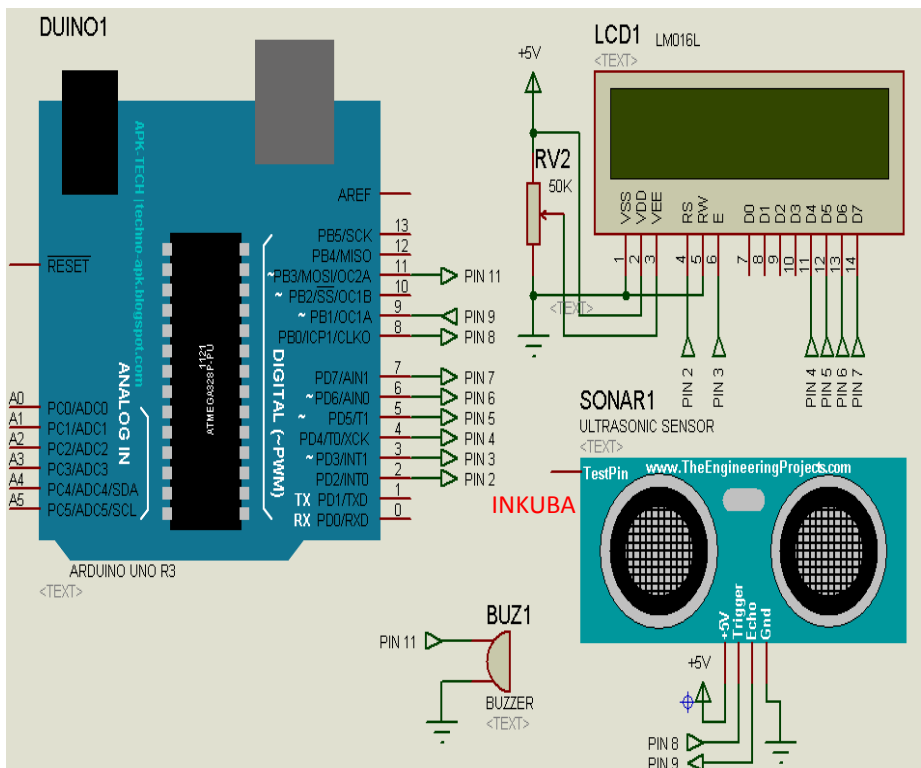
```

5.5. Sensor Parkir Mobil

➤ Komponen :

- Sensor ultrasonic HC – SR04
 - Modul LCD 2 X 16
 - Buzzer
 - Modul Arduino Uno
 - Power Supply ± 9 volt
- Berfungsi untuk men-supply daya

Skematic:



PIN ARDUINO	LCD
2	RS
3	EN
4	D4
5	D5
6	D6
7	D7

Koneksi Sensor Ultrasonic:

PIN SENSOR ULTRASONIC	PIN ARDUINO
VCC	+5V
GND	GND
TRIGER	Pin 8
ECHO	Pin 9

Koneksi Buzzer:

KONEKSI BUZZER	PIN ARDUINO
+BUZZER	Pin 11

➤ **Sistem Kerja Alat :**

Alarm parkir mobil menggunakan sensor jarak yaitu ultrasonic dan arduino sebagai pemrosesnya. Sehingga akan memudahkan kita dalam memperkirakan jarak kendaraan dengan tembok pembatas, dengan cara memberi peringatan berupa suara yang dihasilkan oleh buzzer.

Source Code/ Sketch:

```
#define ECHOPIN 9
#define TRIGPIN 8
#include <LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5,6,7);
int jarak,timer;
void setup(){
  lcd.begin(16, 2);
  lcd.print("  ALARM PARKIR");
  lcd.setCursor(0, 1);
  lcd.print("    MOBIL  ");
  delay(2000);
  lcd.clear();
  lcd.print("Status=");

  pinMode(ECHOPIN, INPUT);
  pinMode(TRIGPIN, OUTPUT);
  pinMode(11,OUTPUT);
```



```

}
void loop(){
  digitalWrite(TRIGPIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGPIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGPIN, LOW);
  timer = pulseIn(ECHOPIN, HIGH);
  jarak= timer/58;
  if (jarak>35){
    lcd.setCursor(7,0);
    lcd.print("Aman  ");
  }
  else if(jarak>30){
    lcd.setCursor(7,0);
    lcd.print("Awasi ");
    digitalWrite(11,1);
    delay(500);
    digitalWrite(11,0);
    delay(500);
  }
  else if(jarak>15){
    lcd.setCursor(7,0);
    lcd.print("Waspada");
    digitalWrite(11,1);
    delay(300);
    digitalWrite(11,0);
    delay(300);
  }
  else {
    lcd.setCursor(7,0);
    lcd.print("Bahaya ");
    digitalWrite(11,1);
  }
}

```

```
    delay(100);  
    digitalWrite(11,0);  
    delay(100);  
  }  
}
```